AD-A064 209 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OHIO SCH--ETC F/G 18/11
APPLICATION OF PHASE-SPACE FINITE ELEMENTS TO THE NEUTRON TRANS--ETC(U)
DEC 78 R C WHEATON

UNCLASSIFIED AFIT/GNE/PH/78D-25 NL

1 OF 1
AD
A064209

END
DATE
FILMED
4-79
DDC

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
(14) AFIT/GNE/PH/78D-25

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) APPLICATION OF PHASE-SPACE FINITE ELEMENTS TO THE NEUTRON TRANSPORT EQUATION IN CYLINDRICAL GEOMETRY.

5. TYPE OF REPORT & PERIOD COVERED
MS Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Ronald C. Wheaton
Captain, USAF

8. CONTRACT OR GRANT NUMBER(s)
(9) Master's thesis,

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology (AFIT-EN)
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

12. REPORT DATE
(11) December 1978

13. NUMBER OF PAGES
69
(12) 69p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

Joseph P. Hipps, Major, USAF JAN 25 1979
Director of Information

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Element Method
Monoenergetic Neutron Transport
Even-parity Transport Equation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Phase-Space finite elements are applied to the static monoenergetic neutron transport equation in two-dimensional cylindrical geometry by computer subroutines which collectively assemble the global phase-space matrix for solution. The technique uses a variational formulation based on the second-order self-adjoint form of the transport equation within which the dependent variable approximated by the finite elements is the even-parity component of the angular flux.

012 225 JOB

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFIT/GNE/PH/78D-25

# APPLICATION OF PHASE-SPACE FINITE ELEMENTS TO THE NEUTRON TRANSPORT EQUATION IN CYLINDRICAL GEOMETRY

THESIS

AFIT/GNE/PH/78D-25

Ronald C. Wheaton

Captain USAF

Approved for public release; distribution unlimited

AFIT/GNE/PH/78D-25

# APPLICATION OF PHASE-SPACE FINITE ELEMENTS TO THE NEUTRON TRANSPORT EQUATION IN CYLINDRICAL GEOMETRY

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Ronald C. Wheaton, B.S.
Captain USAF
Graduate Nuclear Effects
December 1978

Approved for public release; distribution unlimited

# Preface

What began as an evaluation of the angular integrals arising in the finite element solution of the neutron transport equation, has grown into the development of a computational procedure for applying phase-space finite elements to the same equation. The entire project is an extension of the doctoral dissertation work being done by Capt John Souders. As such, I hope that this research answers all of the questions he has posed.

I am grateful to my thesis advisor Lt David D. Hardin, PhD for suggesting this topic and for providing advice, support, constant encouragement, and great patience throughout this endeavor.

I am most appreciative to my wife, Paula, and my daughters, Christine, Amy, and Stacie for their patience and understanding in enduring my many absences while this thesis was being prepared.

Ronald C. Wheaton

## Contents

## List of Figures

AFIT/GNE/PH/78D-25

## Abstract

Phase-Space finite elements are applied to the static monoenergetic neutron transport equation in two-dimensional cylindrical geometry by computer subroutines written by the author to collectively assemble the global phase-space matrix for solution. The technique uses a variational formulation based on the second-order self-adjoint form of the transport equation within which the dependent variable approximated by the finite elements is the even-parity component of the angular flux.

APPLICATION OF PHASE-SPACE FINITE ELEMENTS TO THE
NEUTRON TRANSPORT EQUATION IN CYLINDRICAL GEOMETRY

## I Introduction

The label finite element method was first introduced by Clough in his treatment of plane elasticity problems in 1960, but the initial efforts to use the method had appeared even earlier in the applied mathematics literature with the work of Courant (Ref 1:9). Courant's approach in 1943 used an assemblage of triangular domains with piecewise continuous functions defined over each domain. However, his novel approach did not mature until Greenstadt applied it and developed the mathematical basis for the finite element method as it is used today (Ref 1:10).

Unlike finite difference methods, which lead to a pointwise approximation of the governing differential equations over a solution region that is an array of grid points, the finite element method uses a piecewise approximation to the governing equations for a solution region made up of many small, interconnected sub-regions, or elements. Over each element, the solution of the governing equations is assumed to be both a function of the independent variables and some undetermined coefficients. These element coefficients are then determined so that the assemblage of element solutions is in some sense an optimal approximation to the true solution. In addition, the coefficients are chosen so that any required continuity of the solution or its derivatives is maintained in crossing element boundaries.

By the early 1970's the successful application of the finite element method to problems in solid mechanics, heat conduction, and other areas led to the application of the method to neutronics problems. Initially, the neutron diffusion equations were treated most extensively, followed by increased applications to the neutron transport equation in one-dimensional slab, spherical, and cylindrical geometries as well as two-dimensional cartesian geometry (Refs 2; 3; 4).

Solutions of the transport equation based on finite element techniques often use discrete ordinate approximations for the angular variables (Ref 3). Still other approaches include simultaneous approximations of both the angular and spatial variables by finite elements (Ref 2).

The purpose of this paper is to examine several aspects of the finite element method as applied to the static monoenergetic transport equation with anisotropic scattering and sources in two-dimensional cylindrical geometry. The technique uses a variational formulation based on the second-order self-adjoint form of the transport equation within which the dependent variable approximated by the finite elements is the even-parity component of the angular flux. This component is quite attractive for finite element approximations because it requires that only half of the angular domain be considered. In addition, the even-parity component is always positive and easily integrated to find the scalar flux distribution (Ref 4:149).

Before discussing the computer implementation aspects of the finite element method as applied to the monoenergetic neutron transport equation, the theoretical basis for the second-order even-parity form of the transport equation will be developed. Along with this development, a broad overview of the finite element method will be presented and the variational formulation for the application of the method will be stated. Then the method of computer application will be discussed by describing the computer subroutines written by the author to assemble the system equations for solution. Finally, some conclusions will be presented regarding needed computer code improvements.

# II Theory

As in other fields, the advances in computer technology have been the driving force behind the scientist's ability to solve more and more complex problems as well as the means by which he can increase the accuracy of the problems he has already solved. But advances in computer technology are not the only means of increasing accuracy. Often the application of another numerical method can bring about a significant increase in accuracy. The resulting use of finite elements in solving the neutron transport equation is one of the most recent steps in the quest for increased accuracy.

Present discrete ordinate solution methods lead to anomalous scalar flux distortions when applied to transport problems having strong absorbers and localized neutron sources (Ref 5:255). Thus, as more and more complex geometries are encountered, the finite difference methods can be augmented by the finite element method whose elements can accurately represent complex shapes and eliminate the ray effects. Consequently, computer codes based entirely on the finite element method have been developed to solve the transport equation.

In this chapter the neutron streaming term, $\underline{\Omega}\cdot\nabla\phi$ , of the transport equation will be developed for the problem domain being considered in this thesis. The even-parity form of the transport equation will then be developed, followed by a discussion of the variational formulation of the transport equation and an overview of the finite element method. A brief development of the tensor product basis will conclude the chapter.

## Problem Domain - Cylindrical Coordinates

The monoenergetic transport equation can be written as

$$\underline{\Omega}\cdot\nabla\varphi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\,\varphi(\underline{r},\underline{\Omega}) = \int \sigma_s((\underline{r}),\underline{\Omega}\cdot\underline{\Omega}')\,\varphi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' + S(\underline{r},\underline{\Omega}) \qquad (1)$$

where $\underline{\Omega}\cdot\nabla\varphi$ is the streaming term. In cartesian geometry this streaming term can be calculated quite simply. It has the form

$$\sqrt{1-\mu^2}\left(\cos\chi\,\frac{\partial\Phi}{\partial x} + \sin\chi\,\frac{\partial\Phi}{\partial y}\right) + \mu\,\frac{\partial\Phi}{\partial z} \qquad (2)$$

in rectangular coordinates (Ref 6:59) where $\mu = \underline{\Omega}\cdot\underline{z}$ and $\chi$ is the angle between the planes formed by the $\underline{\Omega}$ and $\underline{z}$ unit vectors and the $\underline{z}$ and $\underline{x}$ unit vectors. However, in curvilinear coordinates the streaming term takes on a somewhat more complicated form. As an example, the $\underline{\Omega}\cdot\nabla\varphi$ term in cylindrical coordinates is (Ref 6:59)

$$\sqrt{1-\mu^2}\cos\chi\,\frac{\partial\Phi}{\partial r} + \frac{\sqrt{1-\mu^2}}{r}\sin\chi\left(\frac{\partial\Phi}{\partial\varphi} - \frac{\partial\Phi}{\partial\chi}\right) + \mu\,\frac{\partial\Phi}{\partial z} \qquad (3)$$

where $\varphi$ is the polar angle, $\mu = \underline{\Omega}\cdot\underline{z}$, and $\chi$ is the angle between the planes formed by the $\underline{\Omega}$ and $\underline{z}$ unit vectors and the $\underline{z}$ and $\underline{r}$ unit vectors. Figure 1 shows the cylindrical coordinate system corresponding to Eq (3).

This paper deals with the application of the finite element method in cylindrical geometry to the transport equation. The problem domain in which cylindrical geometry is assumed is the air over ground burst problem domain (Ref 7) of weapon physics. Since, in this domain the air density varies only with altitude, azimuthal symmetry can be assumed and the conservation form of Eq (2) becomes (Ref 6:58)

$$\frac{\sqrt{1-\mu^2}\cos\chi}{r}\,\frac{\partial(r\Phi)}{\partial r} - \frac{1}{r}\,\frac{\partial\left(\Phi\sqrt{1-\mu^2}\sin\chi\right)}{\partial\chi} + \mu\,\frac{\partial\Phi}{\partial z} \qquad (4)$$

Eq (4) is the streaming term which will be calculated in this thesis by applying the finite element method.

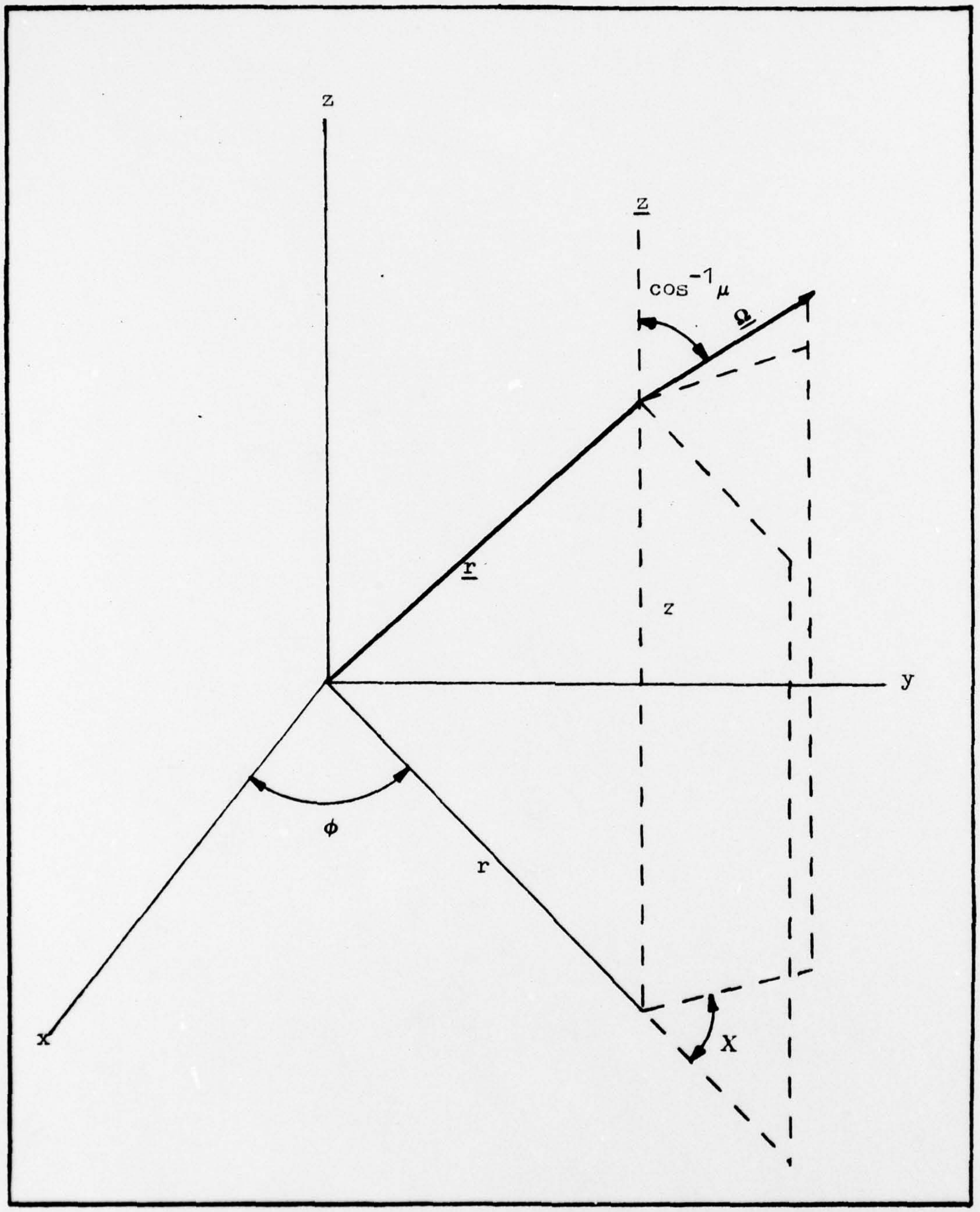

Fig. 1 General Cylindrical Coordinate System

## Even-Parity Transport Equation

The finite element method will be applied to the even-parity form of the transport equation since the even-parity component of the angular flux requires that only half of the angular domain be considered, making it quite attractive for finite element approximations. In addition, the even-parity component is always positive and easily integrated to find the scalar flux distribution (Ref 4:149). The following derivation is that of Kaplan and Davis (Ref 8). Consider the monoenergetic transport equation,

$$\underline{\Omega}\cdot\nabla\varphi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\varphi(\underline{r},\underline{\Omega}) = \int\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\varphi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' + S(\underline{r},\underline{\Omega}) \qquad (1)$$

in the convex region D with total macroscopic cross section $\sigma_t(\underline{r})$ and angular scattering cross section $\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')$ defined such that

$$\sigma_s(\underline{r}) = \int\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,d\underline{\Omega}'$$

where $\sigma_s(\underline{r})$ is the macroscopic scattering cross section. Assume the region is surrounded by a vacuum so that on the surface R of D the vacuum boundary condition is

$$\varphi(\underline{r}_B,\underline{\Omega}) = 0\,, \quad \underline{\Omega}\cdot\underline{n}(\underline{r}_B) < 0 \qquad (5)$$

where $\underline{n}$ is the outward unit normal to R and $\underline{r}_B$ is any $\underline{r} \in$ R. Eq (1) is also valid for $-\underline{\Omega}$ , thus,

$$-\underline{\Omega}\cdot\nabla\varphi(\underline{r},-\underline{\Omega}) + \sigma_t(\underline{r})\varphi(\underline{r},-\underline{\Omega}) = \int\sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}')\varphi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' + S(\underline{r},-\underline{\Omega}) \qquad (6)$$

Adding Eqs (1) and (6) and post-multiplying by 1/2 leads to

$$\begin{aligned}\underline{\Omega}\cdot\nabla\left\{\tfrac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) - \varphi(\underline{r},-\underline{\Omega})\right]\right\} &+ \sigma_t(\underline{r})\left\{\tfrac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) + \varphi(\underline{r},-\underline{\Omega})\right]\right\}\\ &= \int\left\{\tfrac{1}{2}\left[\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') + \sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}')\right]\varphi(\underline{r},\underline{\Omega}')\right\}d\underline{\Omega}'\\ &\qquad + \tfrac{1}{2}\left[S(\underline{r},\underline{\Omega}) + S(\underline{r},-\underline{\Omega})\right] \qquad (7)\end{aligned}$$

or

$$\underline{\Omega}\cdot\nabla\chi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\ \psi(\underline{r},\underline{\Omega}) = \int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\varphi(\underline{r},\underline{\Omega}')d\underline{\Omega}' + S_g(\underline{r},\underline{\Omega}) \quad (8)$$

with the terms $\chi(\underline{r},\underline{\Omega}) = \frac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) - \varphi(\underline{r},-\underline{\Omega})\right]$, $\psi(\underline{r},\underline{\Omega}) = \frac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) + \varphi(\underline{r},-\underline{\Omega})\right]$, $\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\left[\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') + \sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}')\right]$ , and $S_g(\underline{r},\underline{\Omega}) = \frac{1}{2}\left[S(\underline{r},\underline{\Omega}) + S(\underline{r},-\underline{\Omega})\right]$. Let $\chi(\underline{r},\underline{\Omega})$ be the odd-parity flux and $\psi(\underline{r},\underline{\Omega})$ be the even-parity flux. By substracting Eq (6) from Eq (1) and multiplying by 1/2, we get the similar result

$$\begin{aligned}\underline{\Omega}\cdot\nabla\left\{\tfrac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) + \varphi(\underline{r},-\underline{\Omega})\right] + \sigma_t(\underline{r})\ \tfrac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}) - \varphi(\underline{r},-\underline{\Omega})\right]\right\}\\ = \int\left\{\tfrac{1}{2}\left[\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') - \sigma_s((\underline{r}),-\underline{\Omega}\cdot\underline{\Omega}')\right]\varphi(\underline{r},\underline{\Omega}')\right\}d\underline{\Omega}'\\ + \tfrac{1}{2}\left[S(\underline{r},\underline{\Omega}) - S(\underline{r},-\underline{\Omega})\right] \quad (9)\end{aligned}$$

or

$$\underline{\Omega}\cdot\nabla\psi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\ \chi(\underline{r},\underline{\Omega}) = \int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\ \varphi(\underline{r},\underline{\Omega}')d\underline{\Omega}' + S_u(\underline{r},\underline{\Omega}) \quad (10)$$

where the additional substitutions $\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\left[\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') - \sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}')\right]$ and $S_u(\underline{r},\underline{\Omega}) = \frac{1}{2}\left[S(\underline{r},\underline{\Omega}) - S(\underline{r},-\underline{\Omega})\right]$ have been made. Since $\sigma_g$ is an even function of $\underline{\Omega}\cdot\underline{\Omega}'$ , that is, $\sigma_g(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}') = \sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')$ , the integral in Eq (8) can be written as

$$\begin{aligned}\int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\ \varphi(\underline{r},\underline{\Omega}')d\underline{\Omega}' &= \tfrac{1}{2}\int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\varphi(\underline{r},\underline{\Omega}')d\underline{\Omega}' + \tfrac{1}{2}\int\sigma_g(\underline{r},\underline{\Omega}\cdot(-\underline{\Omega}'))\varphi(\underline{r},-\underline{\Omega}')d\underline{\Omega}'\\ &= \int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\ \tfrac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}') + \varphi(\underline{r},-\underline{\Omega}')\right]d\underline{\Omega}'\\ &= \int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\ \psi(\underline{r},\underline{\Omega}')\ d\underline{\Omega}'\end{aligned}$$

and Eq (8) becomes

$$\underline{\Omega}\cdot\nabla\chi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\ \psi(\underline{r},\underline{\Omega}) = \int\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\psi(\underline{r},\underline{\Omega}')d\underline{\Omega}' + S_g(\underline{r},\underline{\Omega}) \quad (11)$$

Similarly, since $\sigma_u$ is an odd function of $\underline{\Omega}\cdot\underline{\Omega}'$ such that $\sigma_u(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}') = -\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')$ , the integral in Eq (10) can be written as

$$\begin{aligned}\int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,\varphi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' &= \frac{1}{2}\int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,\varphi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' + \frac{1}{2}\int\sigma_u(\underline{r},\underline{\Omega}\cdot-\underline{\Omega}')\,\varphi(\underline{r},-\underline{\Omega}')\,d\underline{\Omega}' \\ &= \int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,\frac{1}{2}\left[\varphi(\underline{r},\underline{\Omega}') - \varphi(\underline{r},-\underline{\Omega}')\right]d\underline{\Omega}' \\ &= \int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,\chi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}'\end{aligned}$$

and Eq (10) becomes

$$\underline{\Omega}\cdot\nabla\,\psi(\underline{r},\underline{\Omega}) + \sigma_t(\underline{r})\,\chi(\underline{r},\underline{\Omega}) = \int\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')\,\chi(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' + S_u(\underline{r},\underline{\Omega}) \qquad (12)$$

By putting the boundary condition in terms of the even-parity and odd-parity fluxes, we can write Eq (5) as

$$\begin{aligned}\varphi(\underline{r}_B,\underline{\Omega}) &= \frac{1}{2}\left[\varphi(\underline{r}_B,\underline{\Omega}) + \varphi(\underline{r}_B,-\underline{\Omega})\right] + \frac{1}{2}\left[\varphi(\underline{r}_B,\underline{\Omega}) - \varphi(\underline{r}_B,-\underline{\Omega})\right] \\ &= \psi(\underline{r}_B,\underline{\Omega}) + \chi(\underline{r}_B,\underline{\Omega}) \\ &= 0 \quad , \quad \underline{\Omega}\cdot\underline{n}(\underline{r}_B) < 0 \qquad (13a)\end{aligned}$$

where $\underline{r}_B$ is any $\underline{r} \in$ R and $\underline{n}$ is the outward unit normal to R. In a similar fashion, we can write

$$\varphi(\underline{r}_B,-\underline{\Omega}) = \psi(\underline{r}_B,\underline{\Omega}) - \chi(\underline{r}_B,\underline{\Omega}) = 0 \quad , \quad \underline{\Omega}\cdot\underline{n}(\underline{r}_B) > 0 \qquad (13b)$$

Eqs (11), (12), (13a), and (13b) are the first-order form of the transport equation and its corresponding boundary condition (Ref 8:167).

We now wish to determine the second-order even-parity transport equation in terms of $\psi$, the even-parity flux. In order to do this, we will assume Eq (11) to be an integral equation for $\psi(\underline{r},\underline{\Omega})$ and introduce the linear operator $G_g(\underline{r})$, which maps functions of $\underline{\Omega}$ into other functions of $\underline{\Omega}$ such that for any integrable function, $f(\underline{r},\underline{\Omega})$ ,

$$G_g(\underline{r})\left\{f(\underline{r},\underline{\Omega})\right\} = \sigma_t(\underline{r})\, f(\underline{r},\underline{\Omega}) - \int \sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')f(\underline{r},\underline{\Omega}')\,d\Omega' \qquad (14a)$$

Likewise, introduce the $G_u(\underline{r})$ operator such that

$$G_u(\underline{r})\left[f(\underline{r},\underline{\Omega})\right] = \sigma_t(\underline{r})\, f(\underline{r},\underline{\Omega}) - \int \sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')f(\underline{r},\underline{\Omega}')\,d\underline{\Omega}' \qquad (14b)$$

With the above definitions of the $G_g$ and $G_u$ operators, Eqs (11) and (12) can be rewritten as

$$\underline{\Omega}\cdot\nabla\chi + G_g\psi = S_g$$

$$G_g\psi = S_g - \underline{\Omega}\cdot\nabla\chi \qquad (15a)$$

and

$$\underline{\Omega}\cdot\nabla\psi + G_u\chi = S_u$$

$$G_u\chi = S_u - \underline{\Omega}\cdot\nabla\psi \qquad (15b)$$

For now we will assume that the $G_g$ and $G_u$ operators are invertible and form the new operators $K_g = G_g^{-1}$ and $K_u = G_u^{-1}$. Operating on Eq (15b) with the $G_u^{-1}$ operator, we form

$$G_u^{-1}\left[G_u\chi\right] = G_u^{-1}\left[S_u - \underline{\Omega}\cdot\nabla\psi\right]$$

or

$$\chi = K_u\left[S_u - \underline{\Omega}\cdot\nabla\psi\right] = K_u\left[S_u\right] - K_u\left[\underline{\Omega}\cdot\nabla\psi\right] \qquad (16)$$

Substituting Eq (16) for the odd-parity flux into Eq (15a) results in

$$G_g\psi = S_g - \underline{\Omega}\cdot\nabla\left[K_u S_u\right] + \underline{\Omega}\cdot\nabla\left[K_u\left(\underline{\Omega}\cdot\nabla\psi\right)\right]$$

or

$$-\underline{\Omega}\cdot\nabla\left[K_u\left(\underline{\Omega}\cdot\nabla\psi\right)\right] + G_g\psi = S_g - \underline{\Omega}\cdot\nabla\left[K_u S_u\right] \qquad (17)$$

which is the second-order even-parity form of the transport equation. Its corresponding boundary condition is the dual set

$$\psi(\underline{r},\underline{\Omega}) + K_u[S_u - \underline{\Omega}\cdot\nabla\psi(\underline{r},\underline{\Omega})] = 0 \quad , \quad \underline{\Omega}\cdot\underline{n} < 0 \tag{18a}$$

$$\psi(\underline{r},\underline{\Omega}) - K_u[S_u - \underline{\Omega}\cdot\nabla\psi(\underline{r},\underline{\Omega})] = 0 \quad , \quad \underline{\Omega}\cdot\underline{n} > 0 \tag{18b}$$

Earlier we assumed that the $G_u$ operator was invertible; we will now determine its form in order to make the $K_u$ operator physically significant. The $G_u$ operator defined by Eq (14b) can be proven to be self-adjoint and positive definite (Ref 8:174,175). In forming its inverse, we will take advantage of the fact that $\sigma_u$ depends only on $\underline{\Omega}\cdot\underline{\Omega}'$ and expand $\sigma_s^u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}')$ in Legendre polynomials as

$$\sigma_s^u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{l=0}^{\infty} \sigma_l^u(\underline{r})\left(\frac{2l+1}{4\pi}\right)P_l(\underline{\Omega}\cdot\underline{\Omega}') \tag{19}$$

where $\sigma_s^u$ and $\sigma_l^u$ are macroscopic cross sections as in the previous derivations. Making use of the spherical harmonics addition theorem (Ref 6:609), we can write

$$P_l(\underline{\Omega}\cdot\underline{\Omega}') = \frac{4\pi}{2l+1}\sum_{m=-l}^{+l} Y_{lm}(\underline{\Omega})\, Y_{lm}^{*}(\underline{\Omega}')$$

where * denotes the complex conjugate. Substituting this expression into Eq (19) gives,

$$\sigma_s^u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{l=0}^{\infty}\sum_{m=-l}^{+l} \sigma_l^u(\underline{r})\, Y_{lm}(\underline{\Omega})\, Y_{lm}^{*}(\underline{\Omega}')$$

and Eq (14b) expanded in spherical harmonics assumes the form

$$G_u(\underline{r})[f(\underline{r},\underline{\Omega})] = \sigma_t(\underline{r})\sum_{l=0}^{\infty}\sum_{m=-l}^{+l} f_{lm} Y_{lm}(\underline{\Omega})$$
$$-\int\left[\sum_{l=0}^{\infty}\sigma_l^u\sum_{m=-l}^{+l} Y_{lm}(\underline{\Omega}) Y_{lm}^{*}(\underline{\Omega}')\right]\left[\sum_{j=0}^{\infty}\sum_{k=-j}^{+j} f_{jk} Y_{jk}(\underline{\Omega}')\right] d\underline{\Omega}'$$

Collecting terms, we have

$$G_u(\underline{r})[f(\underline{r},\underline{\Omega})] = \sum_{l=0}^{\infty}\sum_{m=-l}^{+l}\Big[\sigma_t(\underline{r})\, f_{lm}\, Y_{lm}(\underline{\Omega})$$

$$-\sigma_\ell^u Y_{\ell m}(\underline{\Omega})\sum_{j=0}^{\infty}\sum_{k=-j}^{+j} f_{jk}\int Y_{jk}(\underline{\Omega}')Y^*_{\ell m}(\underline{\Omega}')d\underline{\Omega}'$$

or

$$G_u(\underline{r})\left[f(\underline{r},\underline{\Omega})\right] = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\left[\sigma_t f_{\ell m} Y_{\ell m}(\underline{\Omega}) - \sigma_\ell^u Y_{\ell m}(\underline{\Omega})\sum_{j=0}^{\infty}\sum_{k=-j}^{+j} f_{jk}\,\delta_{\ell j}\,\delta_{mk}\right] \tag{20}$$

where $\delta_{lj}$, $\delta_{mk}$ are the Kronecker delta that arise from the fact that the $Y_{lm}$ form a complete orthogonal set (Ref 6:608). Eq (20) can, thus, be written as

$$G_u(\underline{r})\left[f(\underline{r},\underline{\Omega})\right] = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\left[\sigma_t(\underline{r})\, f_{\ell m} Y_{\ell m}(\underline{\Omega}) - \sigma_\ell^u Y_{\ell m}(\underline{\Omega}) f_{\ell m}\right]$$

or

$$G_u(\underline{r})\left[f(\underline{r},\underline{\Omega})\right] = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\left[\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})\right] f_{\ell m} Y_{\ell m}(\underline{\Omega}) \tag{21}$$

The inverse of $G_u(\underline{r})$ is the integral operator

$$K_u\left[f(\underline{r},\underline{\Omega})\right] \equiv G_u^{-1} = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\left(\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})\right)^{-1} f_{\ell m} Y_{\ell m}(\underline{\Omega})$$

or

$$K_u\left[f(\underline{r},\underline{\Omega})\right] = \frac{1}{\sigma_t(\underline{r})}\left[\sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\frac{\sigma_t(\underline{r})}{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})} f_{\ell m} Y_{\ell m}(\underline{\Omega})\right]$$

In the above, the cross section term can be rewritten as

$$\frac{\sigma_t(\underline{r})}{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})} = \frac{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r}) + \sigma_\ell^u(\underline{r})}{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})}$$

$$= 1 + \frac{\sigma_\ell^u(\underline{r})}{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})}$$

so that

$$K_u[f(\underline{r},\underline{\Omega})] = \frac{1}{\sigma_t(\underline{r})}\Big[\sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell} f_{\ell m} Y_{\ell m}(\underline{\Omega})$$

$$+\sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{+\ell}\Big(\frac{\sigma_\ell^u(\underline{r})}{\sigma_t(\underline{r})-\sigma_\ell^u(\underline{r})}\Big) f_{\ell m} Y_{\ell m}(\underline{\Omega})\Big]$$

or

$$K_u[f(\underline{r},\underline{\Omega})] = \frac{1}{\sigma_t(\underline{r})}\Big[ f(\underline{\Omega}) + \int \sigma_{ku}(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') f(\underline{\Omega}') d\underline{\Omega}' \Big] \qquad (22)$$

where

$$\sigma_{ku}(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=0}^{\infty}\frac{2\ell+1}{4\pi}\Big(\frac{\sigma_\ell^u(\underline{r})}{\sigma_t(\underline{r})-\sigma_\ell^u(\underline{r})}\Big) P_\ell(\underline{\Omega}\cdot\underline{\Omega}')$$

is defined as the macroscopic $K_u$ scattering cross section. It will have a finite number of terms whenever the scattering cross section $\sigma_s$ is expanded in a finite number of terms. A similar expression for $K_g$ can also be derived using Eq (14a) and the $\sigma_s^g$ scattering cross section.

Before moving to the section on the finite element method, we will first look at a variational formulation based on the even-parity transport equation which was just derived.

## Variational Formulation

As has been shown, the transport equation and boundary conditions can be derived in various forms. However, a desirable form of the problem is one in which the solution minimizes or makes stationary a functional subject to the given boundary conditions.

It can be proven that the solution of Eq (17) minimizes the functional (Ref 8:169)

$$F(u) = \int_D \Big[ \langle \underline{\Omega}\cdot\nabla u, K_u(\underline{\Omega}\cdot\nabla u)\rangle + \langle u, G_g u\rangle$$

$$-2\langle \underline{\Omega}\cdot\nabla u, K_u S_u\rangle - 2\langle u, S_g\rangle\Big]\, d\underline{r} + \oint_R \Big[\int |\,\underline{\Omega}\cdot\underline{n}(\underline{r})|\, u^2\, d\underline{\Omega}\Big]\, d\underline{r} \qquad (23)$$

where the following definition of the inner product holds on the space of complex functions of $\underline{\Omega}$ :

$$\langle f(\underline{\Omega}), h(\underline{\Omega})\rangle \equiv \int_{\underline{\Omega}} f^*(\underline{\Omega})\, h(\underline{\Omega})\, d\underline{\Omega}$$

and where the second integral is a surface integral over the boundary R of the domain D. For all u satisfying the essential boundary conditions, the variational functional results in the following equivalent <u>weak-form</u> of the second-order even-parity flux equation as derived in Appendix B

$$\begin{aligned}\int_D \Big[\langle \underline{\Omega}\cdot\nabla\eta, K_u(\underline{\Omega}\cdot\nabla\psi)\rangle + \langle \eta, G_g\psi\rangle\Big]\, d\underline{r} \\ + \oint_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}|\,\eta\,\psi\, d\underline{\Omega}\, d\underline{r} = \int_D \Big[\langle \underline{\Omega}\cdot\nabla\eta, K_u S_u\rangle \\ + \langle \eta, S_g\rangle\Big]\, d\underline{r} \qquad (24)\end{aligned}$$

The finite element method will be implemented in the following chapters using this equation for the even-parity flux.

## Finite Element Method

The numerical technique of finite differences uses difference operators to approximate the derivatives in a partial differential equation. In contrast to this, the finite element method does not approximate an operator; it assumes, for an assemblage of discrete elements, a trial solution satisfying the boundary conditions of the problem. The trial solution and its undetermined coefficients are required to satisfy the exact equation in an integral sense; this determines the values of the undetermined coefficients.

Additionally, the accuracy of the problem solution and the degree of approximation depend upon the size and number of elements used and the approximating functions selected for each element. The following concepts will be discussed in this section: the element, generalized coordinates and shape functions, natural coordinates, global nodes, and tent functions.

The Element. In the finite element method the problem domain is divided into a finite number of subdomains, or cells, which are interconnected at nodal points and on the element boundaries. Of course, this division can be quite physical in nature, with each cell being thought of as separate from another like building blocks. Or, the division can be mathematical, with the problem domain (continuum) being zoned into regions by imaginary lines or planes. No matter how the division is done, the finite element method solves the problem collectively for the whole domain by finding a solution for each of its parts, the elements.

Determining the shape of the basic element to be used in the finite element method depends upon such things as the problem geometry and the number of independent variables needed to describe the problem. Thus, one-, two-, and three-dimensional elements with straight or curved sides are possible. Only straight sided elements such as those in Figure 2 will be considered in this thesis.

As stated earlier, the element nodes are points on the element boundary where adjacent elements are considered to be connected. In addition, they are the points where the field variables of the problem

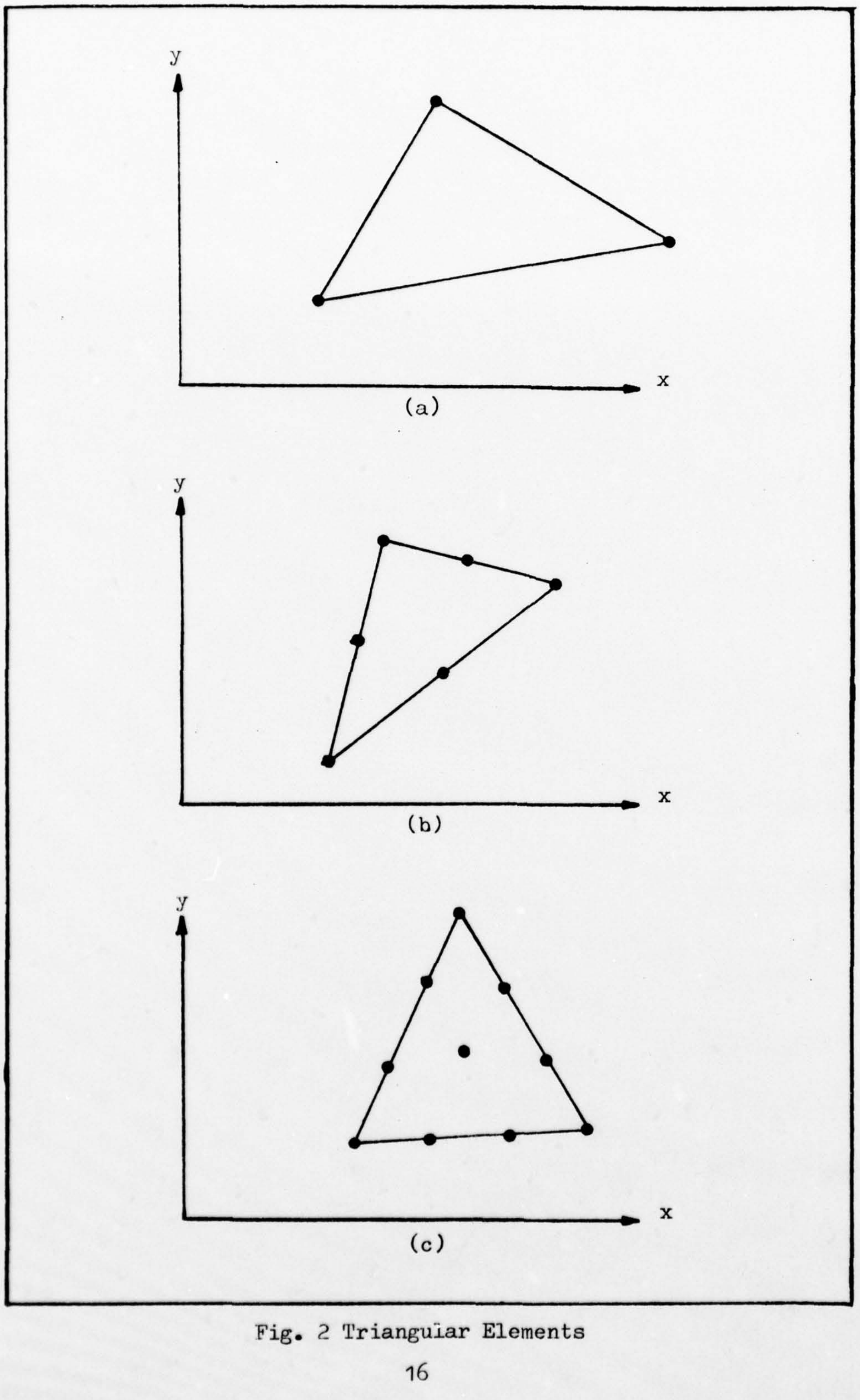

Fig. 2 Trianguiar Elements

are used to define the approximating functions, also known as interpolation functions, for the element. Therefore, nodes can be classified as either exterior (boundary) nodes or as interior nodes (Figure 2c). The element interpolation or shape functions will be discussed in the next section.

Generalized Coordinates and Shape Functions. In the finite element method those functions describing the behavior of the field variables within an element are called approximating functions, interpolation functions, or shape functions (Ref 1:131). Polynomials, which are easily differentiated and integrated as well as mathematically easier to handle in formulating the element equations and in computer calculations, are the most widely used shape functions, although many other functions could be used. Ultimately, the function which is used should obey certain inter-element continuity requirements for the field variable and possibly its derivatives. The additional requirement that the polynomial expansion remain unchanged during a linear transformation from one cartesian coordinate system to another is also desirable. In this paper only two-dimensional polynomials will be used to generate the shape functions. The form of these complete two-dimensional polynomials of order N can be written as (Ref 1:132)

$$P_n(x,y) = \sum_{k=1}^{N_n} \alpha_k x^i y^j, \qquad i + j \leq n$$

where the number of terms in the expansion is $N_n = (n+1)\ (n+2)/2$.

As an example, the shape functions for a rectangular element with nodes at each of its corners will now be derived. Consider the element

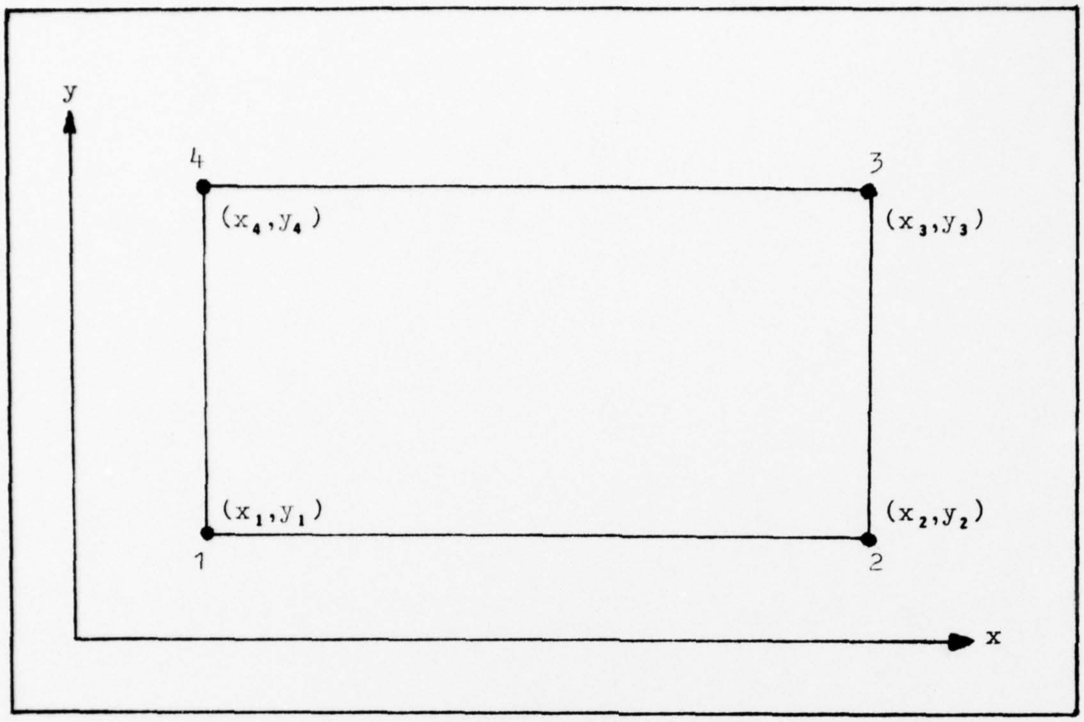

Fig. 3 Rectangular Element

in Figure 3 over which an expression of the following form is assumed:

$$\varphi^{e}(x, y) = \alpha_1^{e} + \alpha_2^{e} x + \alpha_3^{e} y + \alpha_4^{e} x y$$

or

$$\varphi^{e} = [P][\alpha^{e}]^{T} \tag{25}$$

where

$$[\alpha^{e}] = [\alpha_1^{e} \quad \alpha_2^{e} \quad \alpha_3^{e} \quad \alpha_4^{e}]$$

and

$$[P] = [1 \quad x \quad y \quad xy]$$

The superscript e indicates that the equations are only valid for the e$^{th}$ element. The coefficients $\alpha_i$ in this polynomial series representation of the field variable are called generalized coordinates since they have no physical meaning and merely fix the magnitude of the solution $\varphi$. An evaluation of the above expressions at each of the nodes of the rectangle results in the simultaneous equations

$$\begin{aligned}
\varphi_1^e &= \alpha_1^e + \alpha_2^e x_1 + \alpha_3^e y_1 + \alpha_4^e x_1 y_1 \\
\varphi_2^e &= \alpha_1^e + \alpha_2^e x_2 + \alpha_3^e y_2 + \alpha_4^e x_2 y_2 \\
\varphi_3^e &= \alpha_1^e + \alpha_2^e x_3 + \alpha_3^e y_3 + \alpha_4^e x_3 y_3 \\
\varphi_4^e &= \alpha_1^e + \alpha_2^e x_4 + \alpha_3^e y_4 + \alpha_4^e x_4 y_4
\end{aligned}$$

or, in matrix notation,

$$[\varphi^e]^T = [G^e]\,[\alpha^e]^T \tag{26}$$

where

$$[\varphi^e] = [\,\varphi_1^e \quad \varphi_2^e \quad \varphi_3^e \quad \varphi_4^e\,]$$

and

$$[G^e] = \begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 \\ 1 & x_2 & y_2 & x_2y_2 \\ 1 & x_3 & y_3 & x_3y_3 \\ 1 & x_4 & y_4 & x_4y_4 \end{bmatrix}$$

The generalized coordinates can be expressed as the solution of Eq (26), that is,

$$[\alpha^e]^T = [G^e]^{-1}[\varphi^e]^T \tag{27}$$

Substitution of Eq (27) into Eq (25) leads to

$$\varphi^e = [P][G^e]^{-1}[\varphi^e]^T = [N^e][\varphi^e]^T \tag{28a}$$

with

$$[N^e] = [P][G^e]^{-1} \qquad (28b)$$

The $N_i$ are the shape functions associated with the nodal values (or nodal degrees of freedom). As a result, the undetermined coefficients in Eq (28a) now have physical meaning just as do the unknowns in a finite difference scheme. Also, the shape function $N_i$ referring to node i is equal to one at node i and zero at all of the other nodes of the element.

One difficulty encountered in calculating shape functions in this way is the computational effort required to obtain $[G]^{-1}$ when and if it exists. Thus, researchers tried to obtain the shape functions by inspection and, as we shall see, they succeeded with the aid of a special coordinate system called natural coordinates.

Natural Coordinates. In contrast to a global coordinate system which is defined for an entire body or structure, one can define a local coordinate system, called a natural coordinate system, which applies to a specific element (Refs 1:138, 139; 9:88). It is usually set up so that some of the natural coordinates are equal to one at primary external nodes. In this way, the natural coordinates, when used to derive the shape functions, not only simplify the formulation but also facilitate the evaluation of the integrals arising in the element equations. An example of a natural coordinate system for a quadrilateral element (forming a canonical element in the local system) is shown in Figure 4. The rectangular element of Figure 4 is also a member of the serendipity family of elements (Refs 1:170; 9:31-42) which contain only exterior nodes. The shape functions for these

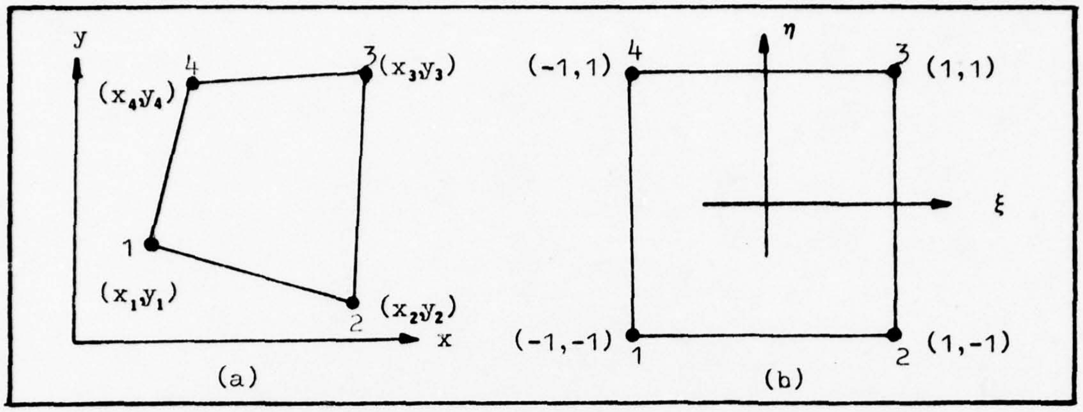

Fig. 4 Natural Coordinates for a General Quadrilateral

elements were found by inspection and have the form

$$N_i(\xi,\eta) = \frac{1}{4}(1+\xi\xi_i)(1+\eta\eta_i) \tag{29}$$

for a linear element (Ref 1:171). The element maintains the necessary continuity of the field variable $\varphi$ along its boundaries.

In concluding this section, it should be noted that the ease with which the shape functions can be found for the rectangular elements makes them appealing for use in the finite element method. But, their use is limited because they cannot represent curved boundaries as well as triangles or elements with curved sides.

Global Nodes and Tent Functions. Once all of the element properties have been found for a system modeled by the finite element method, the overall system properties can be determined by an assembly process. That is, the element matrix equations, Eq (28a), describing the element properties are combined to form the matrix equations describing the properties of the entire modeled system. The assembly process uses nodal compatibility as its basis; the value of the field variable at a node where elements are connected is the same for each element sharing the node. Figure 5 illustrates this point for a node shared by five triangles. In order to implement the assembly process on a digital computer, a global numbering system for the nodes must be formed. Figure 6 shows a mesh of four linear rectangles along with its global node numbering system and each elements local node numbering system. A global node number I is assigned to each node in the mesh. In each element in which a given node appears the index I is used to identify it. For computer use, the global nodes are usually stored in a matrix. Thus, a matrix IE (e,i) may be defined such that IE (e,i) is the global node number of the i$^{th}$ local node in the e$^{th}$ element. The IE matrix

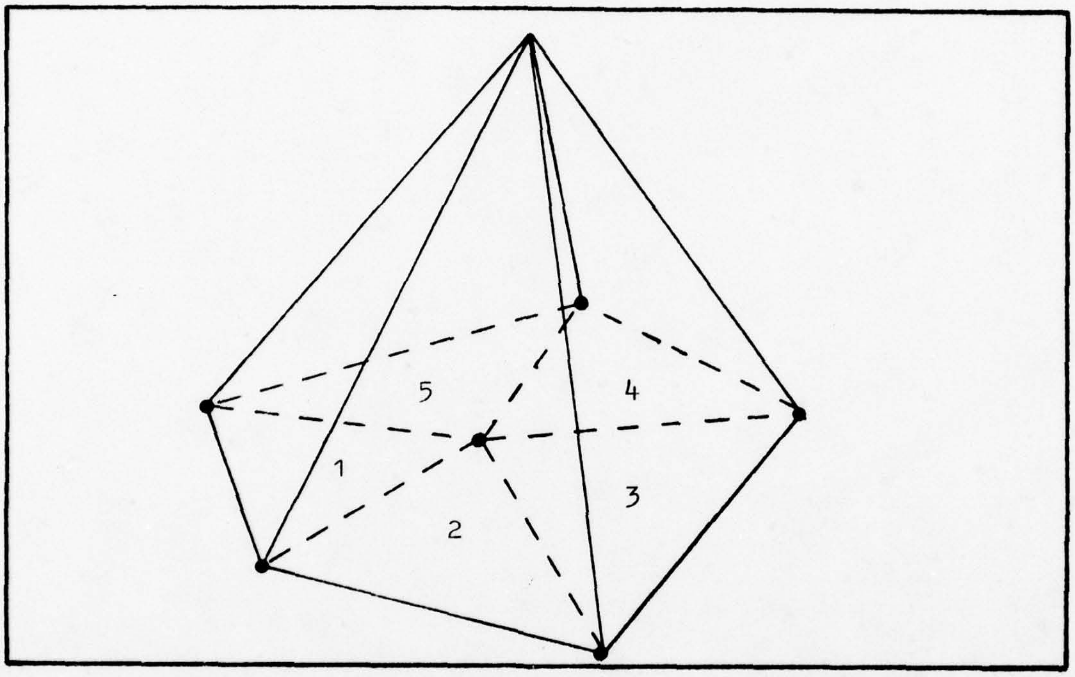

Fig. 5 Tent Function at a Global Node

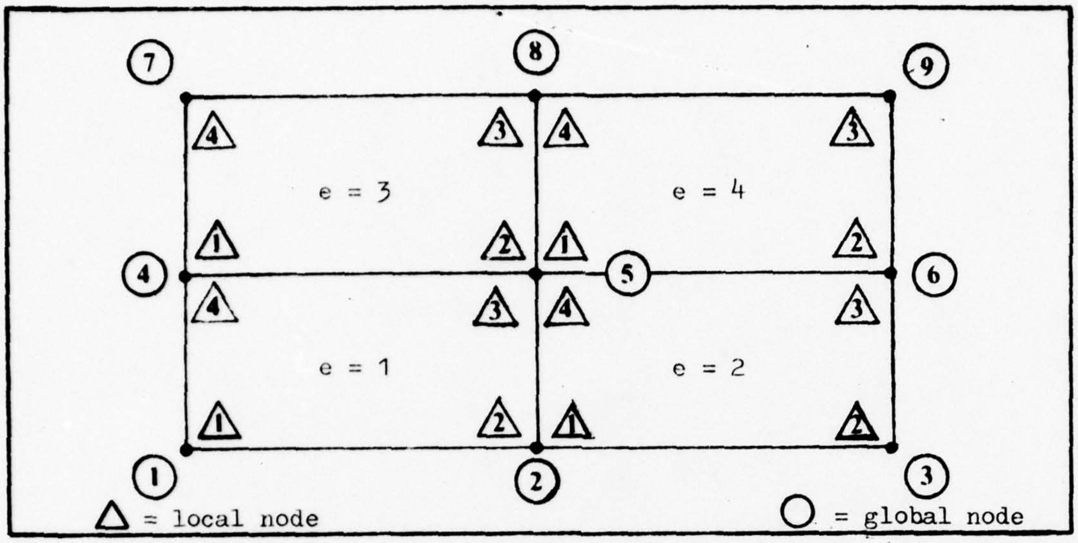

Fig. 6 Finite Element Mesh

for the above rectangular mesh is shown in Figure 7. At each global node in the mesh, a global shape function, or tent function (Fig. 5), is defined as the element shape function of the local node corresponding to the global node. If we denote these tent functions as $T_I(x,y)$

| IE | i = 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| e = 1 | 1 | 2 | 5 | 4 |
| 2 | 2 | 3 | 6 | 5 |
| 3 | 4 | 5 | 8 | 7 |
| 4 | 5 | 6 | 9 | 8 |

Fig. 7 IE Matrix for a Rectangular Element Mesh

and the global node values of the field variable as $\varphi_I$ , then the form of the global problem solution for a mesh of N global nodes is

$$\varphi(x,y) = \sum_{I=1}^{N} \varphi_I \, T_I(x,y) \tag{30}$$

Tensor Product Basis

In this section, the four-dimensional space-angle subspace, within which the finite element method will be applied, is formulated.

In the work of Kaper, Leaf and Lindeman (Ref 2:20), the finite element method is used to solve a six-group transport equation for the tensor product of a finite dimensional subspace whose approximating functions depend on the spatial variables (x,y) and another finite dimensional subspace whose approximating functions depend on the angular variables$(\mu,\eta,\xi)$. In this thesis, a similar tensor product is used, but the spatial and angular subspaces are generated in the cylindrical geometry previously discussed. Thus, the resulting tensor product subspace is four-dimensional with (r,z) spatial variables and$(\mu,\chi)$ angular variables. Mathematically, the formulation can be written as (Ref 10:135)

$$\sum_{i=1}^{N}\sum_{j=1}^{M} S_i(r,z)A_j(\mu,\chi)$$

where

$S_i(r,z)$ = functions of the spatial subspace

$A_j(\mu,\chi)$ = functions of the angular subspace

## III Computer Application

In this chapter the finite element method will be implemented in the cylindrical coordinate system of chapter II using the weak form of the even-parity flux equation

$$\int_D \left[ \langle \underline{\Omega} \cdot \nabla \eta, K_u (\underline{\Omega} \cdot \nabla \psi) \rangle + \langle \eta, G_g \psi \rangle \right] d\underline{r}$$
$$\oint_R \int_{\underline{\Omega}} | \underline{\Omega} \cdot \underline{n} | \, \eta \, \psi \, d\underline{\Omega} \, d\underline{r} = \int_D \left[ \langle \underline{\Omega} \cdot \nabla \eta, K_u S_u \rangle \right.$$
$$\left. + \langle \eta, S_g \rangle \right] d\underline{r} \qquad (24)$$

In addition, it will be assumed that the trial functions in the above formulation, $\psi$ , are equal to $\sum_{i=1}^{N} \sum_{j=1}^{M} q_{ij} \, S_i(r,z) \, A_j(\mu,\chi)$ and that the test functions, $\eta$ , are equal to $A_L(\mu,\chi) \, S_K(r,z)$ where S is the spatial shape function for local nodes i and K and A is the angular shape function for local nodes j and L. These shape functions are calculated by use of Eq (29). By choosing the test functions to be the same as the trial functions used to represent the even-parity flux, the Galerkin method for deriving the finite element equations is established (Ref 1:108).

The implementation of the finite element method in Eq (24) is based on a modular programming approach which uses FORTRAN IV. In this approach the program or code is written in units or modules, each one performing a basic task independent of the other modules. In FORTRAN, these modules are subroutine or function subprograms. In addition, the finite elements used in the formulation of the element equations are the 4-dimensional space-angle phase-space elements of chapter II generated over the tensor product subspace. Since it is difficult to visualize a 4-dimensional element, the subroutines used to code the even-parity equation deal with separate rectangular (r,z) spatial and $(\mu,\chi)$ angular finite elements and their products. These rectangular space and angle finite elements are mapped into the canonical rectangle of Figure 4.

The linear transformation

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = 2\begin{bmatrix} 1/(r_2-r_1) & 1/(z_2-z_1) \end{bmatrix} \begin{bmatrix} r \\ z \end{bmatrix} - \begin{bmatrix} (r_1+r_2)/(r_2-r_1) \\ (z_1+z_2)/(z_2-z_1) \end{bmatrix}$$

takes a rectangle having an arbitrary set of coordinates $(r_1,z_1)$, $(r_2,z_2)$, $(r_3,z_3)$, $(r_4,z_4)$ into the canonical rectangle. A similar transformation is used for the $(\mu,\chi)$ angular element. The inverse mapping is given by

$$\begin{bmatrix} r \\ z \end{bmatrix} = \frac{1}{2}\begin{bmatrix} r_1 \\ z_1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} r_2 \\ z_2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (r_2-r_1) & (z_2-z_1) \end{bmatrix}\begin{bmatrix} \xi \\ \eta \end{bmatrix} \qquad (31)$$

for the spatial element and a similar form is used for the angular element.

Eq (31) and a similar equation for the angular variables were implemented on the computer in such a manner that the coordinate positions in the global spatial and angular elements could be determined immediately after the positions in the corresponding canonical elements had been assigned. This logic was used so that all the element computations could be done for the canonical element with changes to the global variables made as needed. As implemented on the computer, this procedure also permits calculation of the spatial and angular shape functions at local nodes, i,K,j,L in the canonical elements corresponding to the global space-angle elements being considered. In addition, the modular nature of the shape function routine allows either the use of higher order elements or the use of entirely different elements without changing any of the other subroutines.

As stated earlier, the coordinate limits of the canonical element (-1 to +1) make Gaussian numerical integration most attractive for evaluating the space-angle integrals of Eq (24). As a result, Gauss-Legendre quadrature was used in the subroutines. As implemented, the numerical integration is carried out on an element by element basis over the four space-angle variables simultaneously. The integrands calculated in the procedure are the inner product and boundary terms of Eq (24) evaluated for all possible tensor products of the local shape functions. The resulting integrands are thus four-dimensional. Since CDC FORTRAN allows the use of no more than three dimensions in an array, the integrand arrays had to be compacted into two-dimensional arrays in the integration subroutine. Provisions have also been included in the subroutines for use of any combination of up to five user stored Gaussian quadrature rules. In addition to the space-angle integrations, a second integration over the entire angular domain is required by Eqs (14a) and (22) for the $G_g$ and $K_u$ operators.

A closer look at Eqs (14a) and (22) reveals that for every space-angle element being integrated the operators each introduce an additional integral which involves not only the current $\Omega$ neutron direction but also all other possible $\Omega'$ directions. For this reason two separate integration routines were established, one to perform the space-angle integrations element by element and another to perform the angular integrations required by the $G_g$ and $K_u$ operators appearing in the integrands of the space-angle phase-space integrals.

An additional complication also arises when the $G_g$ and $K_u$ operations are performed on a given function of the space-angle variables. Although the same function appears on the right-hand side of Eqs (14a) and (22), outside of the integral as well as within it, the function within the integral depends upon different angular variables. As a result, all functions operated on by the $G_g$ and $K_u$ operators had to be duplicated so that the equations could be properly coded for computer use.

In order to implement the $G_g$ and $K_u$ operations on the computer, it is also necessary to evaluate the scattering cross sections appearing in their definitions. From chapter II we have

$$\sigma_g(\underline{r}, \underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=0}^{\infty} \sigma_\ell^g(\underline{r}) \left(\frac{2\ell+1}{4\pi}\right) P_\ell(\underline{\Omega}\cdot\underline{\Omega}') \tag{32a}$$

$$\sigma_{Ku}(\underline{r}, \underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=0}^{\infty} \frac{2\ell+1}{4\pi} \left(\frac{\sigma_\ell^u(\underline{r})}{\sigma_t(\underline{r}) - \sigma_\ell^u(\underline{r})}\right) P_\ell(\underline{\Omega}\cdot\underline{\Omega}') \tag{32b}$$

where $\sigma_\ell^u$ is the expansion coefficient from Eq (19). In addition, we have

$$\sigma_g(\underline{r}, \underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\left[\sigma_s(\underline{r}, \underline{\Omega}\cdot\underline{\Omega}') + \sigma_s(\underline{r}, -\underline{\Omega}\cdot\underline{\Omega}')\right] \tag{33a}$$

$$\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\left[\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') - \sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}')\right] \qquad (34a)$$

where

$$\sigma_s(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=0}^{\infty} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)P_\ell(\underline{\Omega}\cdot\underline{\Omega}') \qquad (35a)$$

$$\sigma_s(\underline{r},-\underline{\Omega}\cdot\underline{\Omega}) = \sum_{\ell=0}^{\infty} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)P_\ell(-\underline{\Omega}\cdot\underline{\Omega}') \qquad (35b)$$

Substituting Eq (35) into Eqs (33a) and (34a) yields

$$\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\sum_{\ell=0}^{\infty} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)\left[P_\ell(\underline{\Omega}\cdot\underline{\Omega}') + P_\ell(-\underline{\Omega}\cdot\underline{\Omega}')\right]$$

$$\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \frac{1}{2}\sum_{\ell=0}^{\infty} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)\left[P_\ell(\underline{\Omega}\cdot\underline{\Omega}') - P_\ell(-\underline{\Omega}\cdot\underline{\Omega}')\right]$$

or

$$\sigma_g(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=\text{even}}^{L} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)P_\ell(\underline{\Omega}\cdot\underline{\Omega}') \qquad (33b)$$

$$\sigma_u(\underline{r},\underline{\Omega}\cdot\underline{\Omega}') = \sum_{\ell=\text{odd}}^{L} \sigma_\ell^s(\underline{r})\left(\frac{2\ell+1}{4\pi}\right)P_\ell(\underline{\Omega}\cdot\underline{\Omega}') \qquad (34b)$$

Thus, by comparison of Eqs (32a) and (33b) as well as Eqs (19) and (34b), we see that $\sigma_\ell^g = \sigma_\ell^s$ for l even and $\sigma_\ell^u = \sigma_\ell^s$ for l odd. Therefore, all the scattering cross sections can be coded in terms of the known cross sections $\sigma_\ell^s$ which may be either isotropic or anisotropic in nature.

Along with the $G_g$ and $K_u$ operators and their associated scattering cross sections, the $\underline{\Omega}\cdot\nabla$ term of Eq (24) was also coded as developed in chapter II for two-dimensional cylindrical geometry. In contrast, the source terms $S_u$ and $S_g$ as well as the boundary term were not evaluated; they were merely set equal to zero in their respective function subroutines. Several utility subroutines, which were written in order to further modularize the computer coding, brought the total number of subroutines to 45.

The final step in applying the finite element method to the weak form of the second-order even-parity equation is the assembly of the global phase-space matrix from the element phase-space matrices. This procedure is complicated somewhat by the fact that the global space-angle nodes as well as the local space-angle nodes are located in a 4-dimensional phase-space. The assembly subroutine takes this into account by relying on two IE(e,i) matrices, one for the angular domain and one for the spatial domain, to determine the global space and global angle nodes. Once they are calculated, they can be combined to identify the global phase-space node corresponding to a given local spatial node paired with a local angular node.

Once the procedure for labeling the global phase-space nodes has been established, the assembly procedure can be implemented as follows. First, form a null coefficient matrix GA whose dimensions are (# space-angle nodes X # space-angle nodes), or NSAN X NSAN. Then for each phase-space element e = 1,2...,E perform the following steps. Form the element coefficient matrix $\left[ANSF\right]^{e}_{mnxmn}$ , where m = the number of spatial nodes in the element and n = the number of angular nodes in the element, by evaluating Eq (24) for all possible combinations of trial functions, $\psi$ , and test functions, $\eta$ . Find the global phase-space nodes corresponding to the local i, k spatial-j, l angular node pairs. Assemble the local coefficient matrix into its spot in the global coefficient matrix. In abbreviated FORTRAN notation the procedure can be best summarized by the following:

```
C
C           LOOP OVER ELEMENTS
C
        DO 5   e = 1,E
C
C           FORM LOCAL ANS-MATRIX FOR ELEMENT e
C
        CALL GAUSS (e,. . .,ANSF)
C
C           LOOP OVER LOCAL NODES:  GET GLOBAL NODES
C
        DO 4 i = 1,m
        I = IE(e,i)
        DO3 j = 1,n
        J = IE(e,j)
        DO 2 k = 1,m
        K = IE(e,k)
        DO 1 l = 1,n
        L = IE(e,l)
C
C           ASSEMBLE LOCAL INTO GLOBAL
C
        GA(I,J,K,L) = GA(I,J,K,L) + ANSF(i,j,k,l)
      1 CONTINUE
      2 CONTINUE
      3 CONTINUE
      4 CONTINUE
      5 CONTINUE
```

Of course, the source terms would be treated similarly once they were defined in their appropriate subroutines. The accuracy of the global source vector and global coefficient matrix will ultimately depend upon the degree of numerical quadrature used in the finite element integrations. When the global coefficient matrix and global source vector have been formed for the assemblage of elements, the assembly process has thus been completed.

## IV Conclusions and Recommendations

The computer application of the finite element method to the second order even-parity form of the transport equation was partially accomplished in this work by the formation of 45 subroutines which collectively assemble the global phase-space matrix for solution. Because it falls short of actually solving the transport equation, it cannot be considered to be a transport code. Instead, the subroutines generated in this work should be thought of as one of the many approaches possible in applying the method of finite elements. Of course, this approach differs significantly from other similar approaches in that it applies the finite elements in two-dimensional cylindrical geometry and allows the use of anisotropic scattering. Both the cylindrical geometry and the anistropic scattering introduce added complexity to the calculations performed in the subroutines. Yet, the approach is somewhat inefficient. This is due, in part, to the large number of subroutines and to the number of subroutines which had to be duplicated. Inefficiency is also caused by the repeated calculation of all local shape functions for a given spatial or angular element when only one shape function is needed. A revision of this procedure should be considered in future computational refinements and improvements.

As discussed in the last chapter, the four-dimensional nature of the space-angle phase-space finite elements increased the complexity of the digital computations. The doubly subscripted trial and test functions $S_iA_j$ and $S_KA_L$ led to four-dimensional arrays in the local coefficient matrix calculations. These four-dimensional arrays had to be compacted to two dimensions in the subroutines because four-dimensional arrays could not be used in CDC FORTRAN.

Additionally, because of the large number of computations required by the extra angular integrations in the $G_g$ and $K_u$ operations, the subroutines in which they are calculated may use an excessive amount of time. It is recommended that the calculations performed in the operator subroutines as well as the lower level routines utilized by them be streamlined in order to increase their efficiency.

In conclusion, although refinements and improvements in the 45 subroutines of this approach can be made, and development of the source terms and the boundary conditions term accomplished, a method for the application of phase-space finite elements to the anisotropic even-parity neutron transport equation in cylindrical geometry has been presented.

# Bibliography

1. Huebner, Kenneth H. *The Finite Element Method for Engineers*. New York: John Wiley and Sons, Inc., 1975.

2. Kaper, Hans G., *et al*. "Applications of Finite Element Methods in Reactor Mathematics. Numerical Solution of the Neutron Transport Equation," *ANL-8126*. October 1974.

3. Miller, W. F., *et al*. "The Application of Phase-Space Finite Elements to the Two-Dimensional Neutron Transport Equation in X-Y Geometry," *Nuclear Science and Engineering*, *52*: 12-22 (1973).

4. -----. "The Application of Phase-Space Finite Elements to the One-Dimensional Neutron Transport Equation," *Nuclear Science and Engineering*, *51*: 148-156 (1973).

5. Lathrop, K. D. "Remedies for Ray Effects," *Nuclear Science and Engineering*, *45*: 255-268 (1970).

6. Bell, G. I. and S. Glasstone. *Nuclear Reactor Theory*. New York: Van Nostrand Reinhold Co., 1970.

7. Bridgman, C. J. and Richard M. Roberds. "Application of Space-Angle Synthesis to the Two-Dimensional Neutral-Particle Transport Problems of Weapon Physics," *Nuclear Science and Engineering*, *64*: 332-343 (1977).

8. Kaplan, S. and James A. Davis. "Canonical and Involutory Transformations of the Variational Problems of Transport Theory," *Nuclear Science and Engineering*, *28*: 166-176 (1967).

9. Desai, Chandrakant S. and John F. Abel. *Introduction to the Finite Element Method*. New York: Van Nostrand Reinhold Co., 1972.

10. Prenter, P. M. *Splines and Variational Methods*. New York: John Wiley and Sons, Inc., 1975.

# Appendix A

## Computer Subroutines

```
      SUBROUTINE QUAD(KMAX)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE QUAD STORES KMAX GAUSSIAN QUADRATURE RULES ,    *
C WHERE IMAX = THE NUMBER OF QUADRATURE POINTS PER QUADRA-   *
C TURE RULE.     KMAX CANNOT EXCEED 5 .                      *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/ONE/I(5),A(5,50),B(5,50)
      READ*, KMAX
      IF(KMAX.GT.5)GO TO 2
      DO 1 K=1,KMAX
      READ*, I(K)
      IMAX = I(K)
      READ*,((A(K,L),B(K,L)),L=1,IMAX)
    1 CONTINUE
      GO TO 3
    2 PRINT*,"MAXIMUM ALLOWABLE NUMBER OF QUADRATURE RULES
     $IS 5. "
    3 RETURN
      END


      SUBROUTINE QDRULE(IRULE,JRULE)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE QDRULE DETERMINES WHICH OF THE AVAILABLE ( 5    *
C MAXIMUM ) QUADRATURE SETS IS TO BE USED FOR THE SPATIAL    *
C INTEGRATIONS (JPOINT) AND WHICH IS TO BE USED FOR THE      *
C ANGULAR INTEGRATIONS (IPOINT) .                            *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/ONE/I(5),A(5,50),B(5,50)/TWO/C(50),D(50)
      COMMON/THREE/E(50),F(50)/QUADPTS/IPOINT,JPOINT
      READ*, IRULE,JRULE
      IF(IRULE.GT.5.OR.JRULE.GT.5)GO TO 3
      IPOINT = I(IRULE)
      JPOINT = I(JRULE)
      DO 1 K=1,IPOINT
      C(K) = A(IRULE,K)
      D(K) = B(IRULE,K)
    1 CONTINUE
      DO 2 L=1,JPOINT
      E(L) = A(JRULE,L)
      F(L) = B(JRULE,L)
    2 CONTINUE
      GO TO 4
    3 PRINT*,"MAXIMUM NUMBER OF QUADRATURE RULES IS 5 ."
    4 RETURN
      END
      FUNCTION Q(I)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION Q CALCULATES THE QUADRATURE POINTS FOR THE         *
C IPOINT QUADRATURE RULE                                      *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TWO/C(50),D(50)
      Q = C(I)
      RETURN
      END


      FUNCTION W(I)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION W CALCULATES THE QUADRATURE WEIGHTS FOR THE        *
C IPOINT QUADRATURE RULE.                                     *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TWO/C(50),D(50)
      W = D(I)
      RETURN
      END


      FUNCTION QQ(I)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION QQ CALCULATES THE QUADRATURE POINTS FOR THE        *
C JPOINT QUADRATURE RULE.                                     *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/THREE/E(50),F(50)
      QQ = E(I)
      RETURN
      END


      FUNCTION WW(I)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION WW CALCULATES THE QUADRATURE WEIGHTS FOR THE       *
C JPOINT QUADRATURE RULE.                                     *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/THREE/E(50),F(50)
      WW = F(I)
      RETURN
      END
      SUBROUTINE ASSMBL(MAXR,MAXZ,MAXMU,MAXCHI,NAN,NSAN,N,M,
     $     MN,CR,CZ,CAMU,CCHI,IE,JE,SE,AE,GA,GS,ANSF,ANSG)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE ASSMBL DETERMINES THE "STIFFNESS" MATRIX AND     *
C "LOAD" VECTOR FOR EACH SPACE-ANGLE PHASE-SPACE ELEMENT      *
C AND ASSEMBLES BOTH INTO THEIR GLOBAL COUNTERPARTS.          *
C ASSMBL IS CALLED WITH THE MAXIMUM NUMBER OF R,Z,MU, AND     *
C CHI MESH LINES AND THEIR CORRESPONDING ARRAYS, THE VALUES*
C FOR THE NUMBER OF ANGULAR NODES AND THE NUMBER OF SPACE-  *
C ANGLE PHASE-SPACE NODES, THE NUMBER OF LOCAL "SHAPE" FUNC*
C TIONS, AND THE IE & JE GLOBAL NODE MATRICES.                *
C LABELED COMMON BLOCK HILOW IS USED TO TRANSMIT THE ANGU-  *
C LAR MESH ARRAYS TO FUNCTION GAUSS2.                         *
C LABELED COMMON BLOCK ELEMENT PASSES ANGULAR ELEMENT KL TO*
C FUNCTION GAUSS2.                                            *
C       NOMENCLATURE FOR ARGUMENTS....                        *
C             MAXR,MAXZ,MAXMU,MAXCHI = MAXIMUM # OF MESH LINES*
C             NAN = NUMBER OF NODES IN THE ANGULAR DOMAIN     *
C             NSAN = NUMBER OF SPACE-ANGLE PHASE-SPACE NODES  *
C             M = NUMBER OF SPATIAL TENT FUNCTIONS(4=LINEAR,  *
C                 8=QUADRATIC,ETC.).                          *
C             N = NUMBER OF ANGULAR TENT FUNCTIONS            *
C             IE = GLOBAL SPATIAL NODE MATRIX                 *
C             JE = GLOBAL ANGULAR NODE MATRIX                 *
C             SE = SPATIAL ELEMENT NUMBER                     *
C             AE = ANGULAR ELEMENT NUMBER                     *
C             GA = GLOBAL "STIFFNESS" MATRIX                  *
C             GS = GLOBAL "LOAD" VECTOR                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/ELEMENT/KL
      COMMON/HILOW/ALO(50),AHI(50),CHILO(50),CHIHI(50)
      DIMENSION CR(MAXR),CZ(MAXZ),CAMU(MAXMU),CCHI(MAXCHI),
     $    ANSF(MN,MN),ANSG(MN),IE(50,16),JE(50,16),GS(NSAN),
     $    GA(NSAN,NSAN)
      INTEGER SE , AE
C
C INITIALIZE GLOBAL MATRICES.
C
      DO 4 KTEST = 1,M
      K = IE(SE,KTEST)
      DO 3 LTEST=1,N
      L = JE(AE,LTEST)
      LKN = LTEST + (KTEST-1)*N
      KL = L + (K-1)*NAN
      GS(KL) = 0.
      DO 2 ITRIAL =1,M
      I = IE(SE,ITRIAL)
      DO 1 JTRIAL=1,N
      J = JE(AE,JTRIAL)
      JIN = JTRIAL + (ITRIAL-1)*N
      IJ = J + (I-1)*NAN
      GA(KL,IJ) = 0.
    1 CONTINUE
    2 CONTINUE
    3 CONTINUE
    4 CONTINUE
      MAXRM1 = MAXR - 1
      MAXZM1 = MAXZ - 1
      MAXMUM1 = MAXMU - 1
      MAXCHM1 = MAXCHI - 1
      MAXSE = MAXZM1*MAXRM1
      MAXAE = MAXMUM1*MAXCHM1
      CALL HILO(ALO,AHI,CHILO,CHIHI,MAXMU,MAXCHI,MAXMUM1,
     $   MAXCHM1,MAXAE,DAMU,DCHI)
C
C LOOP OVER EACH ELEMENT IN PHASE-SPACE.
C
      DO 12 I=1,MAXZM1
      ZLO = CZ(I)
      ZHI = CZ(I+1)
      DO 11 J=1,MAXRM1
      RLO = CR(J)
      RHI = CR(J+1)
      DO 10 K=1,MAXCHM1
      DO 9 L=1,MAXMUM1
      KL = L + (K-1)*MAXMUM1
C
C     CALCULATE THE ELEMENTAL PHASE-SPACE "STIFFNESS" MATRIX.
C
      CALL GAUSS(N,M,MN,RLO,RHI,ZLO,ZHI,ALO(KL),AHI(KL),
     $   CHIHI(KL),CHILO(KL),ANSF,ANSG)
C
C     LOOP OVER LOCAL NODES, GET GLOBAL NODES.
C
      SE = J + (I-1)*MAXRM1
      AE = KL
      DO 8 KTEST=1,M
      KK = IE(SE,KTEST)
      DO 7 LTEST=1,N
      LL = JE(AE,LTEST)
      LKN = LTEST + (KTEST-1)*N
      KKLL = LL + (KK-1)*NAN
      GS(KKLL) = GS(KKLL) + ANSG(LKN)
      DO 6 ITRIAL=1,M
      II = IE(SE,ITRIAL)
      DO 5 JTRIAL=1,N
      JJ = JE(AE,JTRIAL)
      JIN = JTRIAL + (ITRIAL-1)*N
      IIJJ = JJ + (II-1)*NAN
C
C     ASSEMBLE LOCAL INTO GLOBAL.
C
      GA(KKLL,IIJJ) = GA(KKLL,IIJJ) + ANSF(LKN,JIN)
    5 CONTINUE
    6 CONTINUE
    7 CONTINUE
    8 CONTINUE
    9 CONTINUE
   10 CONTINUE
   11 CONTINUE
   12 CONTINUE
      RETURN
      SUBROUTINE HILO(ALO,AHI,CLO,CHIHI,MAXMU,MAXCHI,MAXMUM1,
     $    MAXCHM1,MAX,CAMU,CCHI)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE HILO CALCULATES THE ALO,AHI,CLO,AND CHIHI VEC-*
C TORS FOR EACH ELEMENT IN THE ANGULAR DOMAIN.              *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DIMENSION ALO(MAX),AHI(MAX),CLO(MAX),CHIHI(MAX),
     $   CAMU(MAXMU),CCHI(MAXCHI)
C
C LOOP OVER EACH ANGULAR ELEMENT AND STORE THE HI AND LO
C VECTORS.
C
      DO 2 I=1,MAXCHM1
      DO 1 J=1,MAXMUM1
      IELEMNT = J + (I-1)*MAXMUM1
      ALO(IELEMNT) = CAMU(J)
      AHI(IELEMNT) = CAMU(J+1)
      CLO(IELEMNT) = CCHI(I)
      CHIHI(IELEMNT) = CCHI(I+1)
    1 CONTINUE
    2 CONTINUE
      RETURN
      END


      SUBROUTINE GAUSS(N,M,MN,RLO,RHI,ZLO,ZHI,ALO,AHI,CHIHI,
     $CHILO,ANSF,ANSG)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C INTEGRATION OF F(R,Z,MU,CHI).DR.DZ.DMU.DCHI BY GAUSSIAN   *
C QUADRATURE.                                                *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      REAL INTGRD1
      COMMON/SPACE/XIK,ETAL/TESTS/KTEST,LTEST
      COMMON/QUADPTS/IPOINT,JPOINT/TRIALS/ITRIAL,JTRIAL
      DIMENSION ANSF(MN,MN),ANSG(MN)

C INITIALIZE ANSF AND ANSG VARIABLES.
C
      DO 1 KTEST =1,M
      DO 1 LTEST=1,N
      LKN = LTEST + (KTEST-1)*N
      ANSG(LKN) = 0.
      DO 1 ITRIAL=1,M
      DO 1 JTRIAL=1,N
      JIN = JTRIAL + (ITRIAL-1)*N
      ANSF(LKN,JIN) = 0.
    1 CONTINUE
C
C     SET UP COORDINATE MAPPING OF ELEMENT IE, I = 1,...,ME.
C
```

```
      CALL MAP(RLO,RHI,RFAC,ZLO,ZHI,ZFAC,ALO,AHI,AFAC,CHILO,
     $         CHIHI,CHIFAC)
      FAC = RFAC * ZFAC * AFAC * CHIFAC
C
C     EVALUATE NEW VARIABLES WHICH ARE EXPRESSED IN TERMS OF
C     THE LEGENDRE ROOTS.
C
      DO 3 I=1,IPOINT
      XII = Q(I)
      DO 3 J=1,IPOINT
      ETAJ = Q(J)
      DO 3 K=1,JPOINT
      XIK = QQ(K)
      DO 3 L=1,JPOINT
      ETAL = QQ(L)
      WEIGHT = W(I)*W(J)*WW(K)*WW(L)
C
C     SET UP TERMS TO BE INTEGRATED AND CARRY OUT INTEGRATION
C     BY CALCULATING      ANS = SUM OF W*F(XI,ETA)  .
C
      DO 2 KTEST=1,M
      DO 2 LTEST=1,N
      LKN = LTEST  + (KTEST-1)*N
      ANSG(LKN) = ANSG(LKN)  + WEIGHT*INTGRD2(XIK,ETAL,XII,
     $    ETAJ)*FAC
      DO 2 ITRIAL =1,M
      DO 2 JTRIAL =1,N
      JIN =  JTRIAL  + (ITRIAL-1)*N
      ANSF(LKN,JIN) = ANSF(LKN,JIN)  + WEIGHT*INTGRD1(XIK,
     $    ETAL,XII,ETAJ)*FAC
    2 CONTINUE
    3 CONTINUE
      RETURN
      END

      SUBROUTINE MAP(WLO,WHI,WFAC,XLO,XHI,XFAC,YLO,YHI,YFAC,
     $               ZLO,ZHI,ZFAC)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE COMPUTES THE SCALING FACTORS WFAC,XFAC,     *
C YFAC, AND ZFAC USED TO MAP COORDINATES IN THE W,X & Y,Z     *
C PLANES INTO THE W',X' & Y',Z' PLANES.                       *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPSPAC/A,B,C,D/MAPANGL/E,F,G,H
      A = (WLO + WHI)/2.
      B = (WHI - WLO)/2.
      C = (XHI + XLO)/2.
      D = (XHI - XLO)/2.
      E = (YHI + YLO)/2.
      F = (YHI - YLO)/2.
      G = (ZHI + ZLO)/2.
      H = (ZHI - ZLO)/2.
      WFAC = B
      XFAC = D
      YFAC = F
      ZFAC = H
      RETURN
      END


      REAL FUNCTION INTGRD1(XIK,ETAL,XII,ETAJ)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE INTGRD1 CALCULATES THE INTEGRAND OF THE STREAM*
C ING AND SCATTERING PHASE SPACE MATRIX IN THE WEAK FORM OF*
C THE "EVEN"-PARITY SECOND ORDER TRANSPORT EQUATION.       *
C SUBROUTINE INTGRD1 USES THE FOLLOWING FUNCTION SUB/R'S   *
C               RKU           ODELTST       BNDCOS         *
C               GG            ODELTRL       ODELTR2        *
C               TRIAL         TEST          TRIAL2         *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      EXTERNAL ODELTRL,ODELTR2,TRIAL,TRIAL2
      F1 = RKU(ODELTRL,ODELTR2,XIK,ETAL,XII,ETAJ)*ODELTST
     $   (XIK,ETAL,XII,ETAJ)
      F2 = GG(TRIAL,TRIAL2,XIK,ETAL,XII,ETAJ)*TEST(XIK,ETAL,
     $   XII,ETAJ)
      F3 = TRIAL(XIK,ETAL,XII,ETAJ)*TEST(XIK,ETAL,XII,ETAJ)
     $   *BNDCOS(XIK,ETAL,XII,ETAJ)
      INTGRD1 = F1 + F2 + F3
      RETURN
      END


      FUNCTION RKU(FCT,FCT2,XIK,ETAL,XII,ETAJ)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION RKU DETERMINES THE VALUE OF THE OPERATOR KU.    *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      EXTERNAL SIGMAKU,FCT2
      RK = R(XIK)
      ZL = Z(ETAL)
      AMUI = AMU(XII)
      CHIJ = CHI(ETAJ)
      RKU = FCT(XIK,ETAL,XII,ETAJ)/XT(RK,ZL) +
     $   GAUSS2(SIGMAKU,FCT2,RK,ZL,AMUI,CHIJ)
      RETURN
      END
      REAL FUNCTION INTGRD2(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C SUBROUTINE INTGRD2 CALCULATES THE INTEGRANDS OF THE          *
C SOURCE VECTOR IN THE WEAK FORM OF THE "EVEN"-PARITY SECOND
C ORDER TRANSPORT EQUATION.                                     *
C SUBROUTINE INTGRD2 USES THE FOLLOWING FUNCTION SUB/R'S       *
C                     SG             TEST                       *
C                     RKU            ODELTST                    *
C                              SU                               *
C FUNCTION SU WHEN OPERATED ON BY FUNCTION RKU WILL REQUIRE*
C MODIFICATION (SEE FUNCTIONS ODELTR2 AND TRIAL2).              *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      EXTERNAL SU
      F1 = SG(XIK,ETAL,XII,ETAJ)*TEST(XIK,ETAL,XII,ETAJ)
      F2 = RKU(SU,XIK,ETAL,XII,ETAJ)*ODELTST(XIK,ETAL,XII,
     $   ETAJ)
      INTGRD2 = F1 + F2
      RETURN
      END


      FUNCTION GG(FCT,FCT2,XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION GG DETERMINES THE VALUE OF THE OPERATOR GG.          *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      EXTERNAL SIGMAG,FCT2
      RK = R(XIK)
      ZL = Z(ETAL)
      AMUI = AMU(XII)
      CHIJ = CHI(ETAJ)
      GG = XT(RK,ZL)*FCT(XIK,ETAL,XII,ETAJ)  - GAUSS2(SIGMAG,
     $   FCT2,RK,ZL,AMUI,CHIJ)
      RETURN
      END


      FUNCTION XT(R,Z)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION XT CALCULATES THE ANISOTROPIC TOTAL CROSS SECT. *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      XT = 1.
      END
      FUNCTION ODELTRL(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION ODELTRL CALCULATES THE OMEGA DOT DEL TRIAL TERM *
C (THE STREAMING TERM) IN CYLINDRICAL COORDINATES.          *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/NODES/M,N/TRIALS/I,J
      TR = TRIAL(XIK,ETAL,XII,ETAJ)
      CALL DERIV(M,N,I,J,D1,D2,D3)
      RK = R(XIK)
      AMUI = AMU(XII)
      CHIJ = CHI(ETAJ)
      A = 1. - AMUI**2
      B = SQRT(A) / RK
      ODELTRL = B*COS(CHIJ)*((RK*D1)+TR) - B*D2 + AMUI*D3
      RETURN
      END



      FUNCTION ODELTR2(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION ODELTR2 CALCULATES THE OMEGA DOT DEL TRIAL TERM *
C (THE STREAMING TERM) IN CYLINDRICAL COORDINATES FOR USE  *
C IN FUNCTION GAUSS2.                                       *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/NODES/M,N/TRIALS/I,J/LNODE/JJ
      TR2 = TRIAL2(XIK,ETAL,XII,ETAJ)
      CALL DERIV2(M,N,I,JJ,D1,D2,D3)
      RK = R(XIK)
      AMUI = AMU(XII)
      CHIJ = CHI(ETAJ)
      A = 1. - AMUI**2
      B = SQRT(A) / RK
      ODELTR2 = B*COS(CHIJ)*((RK*D1) + TR2) - B*D2 + AMU1*D3
      RETURN
      END



      FUNCTION ODELTST(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION ODELTST CALCULATES THE OMEGA DOT DEL TEST TERM  *
C (THE STREAMING TERM) IN CYLINDRICAL COORDINATES.          *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
```

```
      COMMON/NODES/I,J/TESTS/K,L
      TST = TEST(XIK,ETAL,XII,ETAJ)
      CALL DERIV(M,N,K,L,D1,D2,D3)
      RK = R(XIK)
      AMUI = AMU(XII)
      CHIJ = CHI(ETAJ)
      A = 1. - AMUI**2
      B = SQRT(A) / RK
      ODELTST = B*COS(CHIJ)*((RK*D1) + TST) - B*D2 + AMUI*D3
      RETURN
      END


      FUNCTION TRIAL(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION TRIAL CALCULATES THE VALUE OF THE TRIAL SOLUTION*
C S(I)*A(J).                                                       *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TRIALS/I,J
      COMMON/TENT1/S,DTDXI1,DTDETA1/TENT2/A,DTDXI2,DTDETA2
      DIMENSION S(16),A(16),DTDXI1(16),DTDXI2(16),
     $          DTDETA1(16),DTDETA2(16)
      CALL TENTFNC(XIK,ETAL,S,DTDXI1,DTDETA1)
      CALL TENTFNC(XII,ETAJ,A,DTDXI2,DTDETA2)
      TRIAL = S(I)*A(J)
      RETURN
      END


      SUBROUTINE SCALE(N,DTDXI,DTDETA,AMUFAC,CHIFAC,DTDCHI,
     $                  DTDMU,T,CHI,XSIN)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE CALCULATES, FOR A GIVEN ELEMENT E,THE          *
C DERIVATIVE OF THE "TEST FUNCTIONS", T(I), WITH RESPECT TO*
C CHI AND MU BY SCALING THE VALUES DTDXI AND DTDETA BY           *
C (DETA/DCHI) = 1./CHIFAC AND (DXI/DMU) = 1./AMUFAC.  AD-        *
C DITIONALLY, XSIN = THE PARTIAL DERIVATIVE OF T(I) *            *
C SIN(CHI) WITH RESPECT TO CHI IS COMPUTED.                      *
C     AMUFAC = SCALING FACTOR SUPPLIED BY SUBROUTINE MAP.        *
C     CHIFAC = SCALING FACTOR SUPPLIED BY SUBROUTINE MAP.        *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DIMENSION DTDXI(N),DTDETA(N),DTDMU(N),DTDCHI(N),T(N),
     $          XSIN(N)
      DO 1 I=1,N
      DTDCHI(I)=DTDETA(I)*(1./CHIFAC)
      DTDMU(I)=DTDXI(I)*(1./AMUFAC)
      XSIN(I)=DTDCHI(I)*SIN(CHI)+T(I)*COS(CHI)
    1 CONTINUE
      RETURN
      END
      SUBROUTINE TENTFNC (XI,ETA,T,DTDXI,DTDETA)
      DIMENSION T(4),DTDXI(4),DTDETA(4)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE CALCULATES THE VALUES FOR THE TENT FUNC- *
C TIONS AND THEIR DERIVATIVES ON THE BASIS OF A CANONICAL  *
C ELEMENT WHICH IS A LINEAR SERENDIPITY RECTANGLE WITH     *
C NODES AT (XI,ETA) = (-1,-1),(1,-1),(1,1), AND (-1,1).    *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C CALCULATE THE TENT FUNCTIONS T(I).
C
      T(1)=(1.-XI)*(1.-ETA)/4.
      T(2)=(1.+XI)*(1.-ETA)/4.
      T(3)=(1.+XI)*(1.+ETA)/4.
      T(4)=(1.-XI)*(1.+ETA)/4.
C
C CALCULATE THE DERIVATIVES OF THE T(I) WITH RESPECT TO XI.
C
      DTDXI(1)=(ETA-1.)/4.
      DTDXI(2)=-DTDXI(1)
      DTDXI(3)=(1.+ETA)/4.
      DTDXI(4)=-DTDXI(3)
C
C CALCULATE THE DERIVATIVES OF THE T(I) WITH RESPECT TO ETA.
C
      DTDETA(1)=(XI-1.)/4.
      DTDETA(2)=(-1.-XI)/4.
      DTDETA(3)=-DTDETA(2)
      DTDETA(4)=-DTDETA(1)
      RETURN
      END


      FUNCTION TEST(XIK,ETAL,XII,ETAJ)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION TEST CALCULATES THE VALUE OF THE TEST SOLUTION  *
C S(K)*A(L).                                                *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TESTS/K,L
      COMMON/TENT1/S,DTDXI1,DTDETA1/TENT2/A,DTDXI2,DTDETA2
      DIMENSION S(16),A(16),DTDXI1(16),DTDXI2(16),
     $          DTDETA1(16),DTDETA2(16)
      CALL TENTFNC(XIK,ETAL,S,DTDXI1,DTDETA1)
      CALL TENTFNC(XII,ETAJ,A,DTDXI2,DTDETA2)
      TEST = S(K)*A(L)
      RETURN
      END
      SUBROUTINE DERIV(M,N,I,J,D1,D2,D3)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C       D1 CALCULATES THE VALUE PARTIAL(R*FCT)/PARTIAL(R)      *
C APPEARING IN THE STREAMING TERM.                             *
C       D2 CALCULATES THE VALUE PARTIAL(FCT*SIN(CHI))/PARTIAL*
C CHI APPEARING IN THE STREAMING TERM.                         *
C       D3 CALCULATES THE VALUE PARTIAL(FCT)/PARTIAL(Z) AP-    *
C PEARING IN THE STREAMING TERM.                               *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TENT1/S,DSDXI,DSDETA/TENT2/A,DADXI,DADETA
      COMMON/MAPSPAC/AA,RFAC,C,ZFAC/MAPANGL/E,AFAC,G,CFAC
      DIMENSION DSDXI(16),DSDETA(16),DSDZ(16),DSDR(16),
     $    DSDSIN(16),DADXI(16),DADETA(16),DADCHI(16),S(16),
     $    A(16),DADMU(16),DADSIN(16)
      CALL SCALE(M,DSDXI,DSDETA,RFAC,ZFAC,DSDZ,DSDR,S,Z,
     $    DSDSIN)
      CALL SCALE(N,DADXI,DADETA,AFAC,CFAC,DADCHI,DADMU,
     $    A.CHI,DADSIN)
      D1 = A(J)*DSDR(I)
      D2 = S(I)*DADSIN(J)
      D3 = A(J)*DSDZ(I)
      RETURN
      END


      SUBROUTINE DERIV2(M,N,I,J,D1,D2,D3)
C
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE IS USED BY FUNCTION DDELTR2.                 *
C       D1 CALCULATES THE VALUE PARTIAL(R*FCT)/PARTIAL(R)      *
C APPEARING IN THE STREAMING TERM.                             *
C       D2 CALCULATES THE VALUE PARTIAL(FCT*SIN(CHI))/PARTIAL*
C CHI APPEARING IN THE STREAMING TERM.                         *
C       D3 CALCULATES THE VALUE PARTIAL(FCT)/PARTIAL(Z) AP-    *
C PEARING IN THE STREAMING TERM.                               *
C*   * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TENT1/S,DSDXI,DSDETA/TENTTWO/A,DADXI,DADETA
      COMMON/MAPSPAC/AA,RFAC,C,ZFAC/MAPANG2/X,AFAC,Y,CFAC
      DIMENSION DSDXI(16),DSDETA(16),DSDZ(16),DSDR(16),
     $    DSDSIN(16),DADXI(16),DADETA(16),DADCHI(16),S(16),
     $    A(16),DADMU(16),DADSIN(16)
      CALL SCALE(M,DSDXI,DSDETA,RFAC,ZFAC,DSDZ,DSDR,S,Z,
     $    DSDSIN)
      CALL SCALE(N,DADXI,DADETA,AFAC,CFAC,DADCHI,DADMU,
     $    A,CHI,DADSIN)
      D1 = A(J)*DSDR(I)
      D2 = S(I)*DADSIN(J)
      D3 = A(J)*DSDZ(I)
      RETURN
      END
      FUNCTION TRIAL2(XIK,ETAL,XII,ETAJ)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION TRIAL 2 CALCULATES THE VALUE OF THE TRIAL SOLU- *
C TION S(I)*A(LOCAL) TO BE NUMERICALLY INTEGRATED IN       *
C FUNCTION GAUSS2.                                         *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/TRIALS/I,J/LNODE/LOCAL
      COMMON/TENT1/S,DTDXI1,DTDETA1/TENTTWO/A,DTDXI2,DTDETA2
      DIMENSION S(16),A(16),DTDXI1(16),DTDXI2(16),
     $          DTDETA1(16),DTDETA2(16)
      CALL TENTFNC(XIK,ETAL,S,DTDXI1,DTDETA1)
      CALL TENTFNC(XII,ETAJ,A,DTDXI2,DTDETA2)
      TRIAL2 = S(I)*A(LOCAL)
      RETURN
      END


      FUNCTION SIGMAKU(R,Z,AMUI,CHIJ,AMUK,CHIL)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS FUNCTION EVALUATES THE "KU" MACROSCOPIC CROSS SECT. *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AMUNOT = UZERO(AMUI,CHIJ,AMUK,CHIL)
      PI = 3.1415926535
      SIGMAKU = 0.
      L = LFORPL(IDUMMY)
      IF(L.EQ.0)RETURN
      DO 1 I=1,L,2
      SIGMAKU = SIGMAKU + (2*I+1)*XS(R,Z,I)*POLY(I,AMUNOT)
     $   /4./PI/(XT(R,Z)-XS(R,Z,I))
    1 CONTINUE
      RETURN
      END


      FUNCTION UZERO(U,X,UPRIME,XPRIME)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION UZERO COMPUTES THE VALUE MU-ZERO FOR THE TERM   *
C OMEGA DOT OMEGA PRIME IN CYLINDRICAL COORDINATES.        *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      Z1 = 1. - U**2
      Z2 = 1. - UPRIME**2
      DX = X - XPRIME
      UZERO = U * UPRIME + SQRT(Z1) * SQRT(Z2) * COS(DX)
      RETURN
      END
      FUNCTION SIGMAG(R,Z,AMUI,CHIJ,AMUK,CHIL)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS FUNCTION DETERMINES THE VALUE OF THE "EVEN" MACRO-    *
C SCOPIC SCATTERING CROSS SECTION.                            *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      AMUNOT = UZERO(AMUI,CHIJ,AMUK,CHIL)
      PI = 3.1415926535
      SIGMAG = XS(R,Z,0)/4./PI
      L = LFORPL(IDUMMY)
      IF(L.LE.1)RETURN
      DO 1 I=2,L,2
      SIGMAG = SIGMAG + (2*I+1)*XS(R,Z,I)*POLY(I,AMUNOT)
     $   /4./PI
    1 CONTINUE
      RETURN
      END


      FUNCTION LFORPL(IDUMMY)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION LFORPL CALCULATES THE ORDER OF THE CROSS SECTION*
C EXPANSIONS.                                                 *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/EXPAND/IPL
      LFORPL = IPL
      RETURN
      END


      FUNCTION XS(R,Z,L)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION XS CALCULATES THE ANISOTROPIC SCATTERING CROSS    *
C SECTION.                                                    *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      XS = 1.
      IF(L.GT.0)XS=0.
      RETURN
      END
      FUNCTION POLY(I,X)
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE CALCULATES THE I'TH LEGENDRE POLYNOMIAL   *
C FOR AN INPUT VALUE OF X.                                   *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      POLY = 1.
      IF(I.EQ.0)RETURN
      POLY = X
      IF(I.EQ.1) RETURN
      PIM1 = 1.
      PI = X
      DO 1 L=2,I
      POLY = ((2*L-1)*X*PI-(L-1)*PIM1)/L
      PIM1 = PI
      PI = POLY
    1 CONTINUE
      RETURN
      END


      FUNCTION SG(R,Z,AMU,CHI)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION SG DETERMINES THE VALUE OF THE "EVEN"-PARITY     *
C SOURCE TERM.                                              *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SG = 0.
      RETURN
      END


      FUNCTION SU(R,Z,AMU,CHI)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION SU DETERMINES THE VALUE OF THE "ODD"-PARITY      *
C SOURCE TERM.                                              *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      SU = 0.
      RETURN
      END


      FUNCTION BNDCOS(R,Z,AMU,CHI)
C
C*  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION BNDCOS CALCULATES THE BOUNDARY TERMS FOR THE EL-*
C EMENT.                                                    *
C*  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      BNDCOS = 0.
      RETURN
      END
      FUNCTION R(XI)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION R CALCULATES THE RADIAL COORDINATE CORRESPONDING*
C TO THE NON-DIMENSIONAL XI COORDINATE OF THE "CANONICAL"  *
C ELEMENT.                                                 *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPSPAC/A,B,C,D
      R = A + B*XI
      RETURN
      END


      FUNCTION Z(ETA)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION Z CALCULATES THE Z COORDINATE CORRESPONDING TO  *
C THE NON-DIMENSIONAL ETA COORDINATE OF THE "CANONICAL"    *
C ELEMENT.                                                 *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPSPAC/A,B,C,D
      Z = C + D*ETA
      RETURN
      END


      FUNCTION AMU(XI)
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C FUNCTION AMU CALCULATES THE MU COORDINATE CORRESPONDING  *
C TO THE NON-DIMENSIONAL XI COORDINATE OF THE "CANONICAL"  *
C ELEMENT.                                                 *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPANGL/E,F,G,H
      AMU = E + F*XI
      RETURN
      END


      FUNCTION CHI(ETA)
C
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION CHI CALCULATES THE CHI COORDINATE CORRESPONDING *
TO THE NON-DIMENSIONAL ETA COORDINATE OF THE "CANONICAL"   *
C ELEMENT.                                                 *
C*    * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPANGL/E,F,G,H
      CHI = G + H*ETA
      RETURN
      END
      SUBROUTINE MAP2(ALO,AHI,AFAC,CLO,CHI,CFAC)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE COMPUTES THE SCALING FACTORS AFAC AND     *
C CFAC AS WELL AS THE COORDINATE MAPPING VARIABLES USED TO *
C MAP COORDINATES IN THE MU,CHI ANGULAR PLANE INTO THE XI, *
C ETA "CANONICAL" PLANE.  THIS ROUTINE IS USED BY FUNCTION *
C GAUSS2.                                                  *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPANG2/A,B,C,D
      A = (AHI + ALO)/2.
      B = (AHI - ALO)/2.
      C = (CHI + CLO)/2.
      D = (CHI - CLO)/2.
      AFAC = B
      CFAC = D
      RETURN
      END


      FUNCTION CHI2(ETA)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION CHI2 CALCULATES THE CHI COORDINATE CORRESPONDING*
C TO THE ETA COORDINATE OF THE "CANONICAL" ELEMENT.        *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPANG2/A,B,C,D
      CHI2 = C + D*ETA
      RETURN
      END


      FUNCTION AMU2(XI)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C FUNCTION AMU2 CALCULATES THE MU COORDINATE CORRESPONDING *
C TO THE NON-DIMENSIONAL XI COORDINATE OF THE "CANONICAL"  *
C ELEMENT.                                                 *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/MAPANG2/A,B
      AMU2 = A + B*XI
      RETURN
      END
      FUNCTION GAUSS2(FCT1,FCT2,R,Z,AMU,CHI)
C
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C GAUSS2 IS A GAUSS-LEGENDRE MULTIPLE INTEGRATION ROUTINE         *
C WHICH USES AN IPOINT QUADRATURE RULE OVER THE MU,CHI            *
C VARIABLES.                                                      *
C                                                                 *
C JE(E,JTRIAL) IS THE GLOBAL ANGLE NODE MATRIX.                   *
C AFAC AND CFAC ARE COORDINATE SCALING FACTORS SUPPLIED BY        *
C SUBROUTINE MAP2.                                                *
C                                                                 *
C FUNCTION GAUSS2 USES FUNCTIONS AMU2,CHI2,Q,W,FCT1,FCT2,         *
C NE,KE, AND LNN.                                                 *
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      COMMON/HILOW/ALO(50),AHI(50),CLO(50),CHIHI(50)
      COMMON/SPACE/XIK,ETAL/LNODE/LOCAL/QUADPTS/IPOINT
      COMMON/NODES/M,N/ELEMENT/E/ANGELM/NANGEL
      COMMON/TRIALS/ITRIAL,JTRIAL/JEMAT/JE(50,16)
      INTEGER E
      GAUSS2 = 0.
C
C DETERMINE THE GLOBAL ANGLE NODE OF THE LOCAL NODE JTRIAL
C USED IN THE OUTER ANGULAR ELEMENT LOOP.
C
      J = JE(E,JTRIAL)
C
C DETERMINE THE NUMBER OF ANGULAR ELEMENTS WHICH CONTAIN
C GLOBAL NODE J.
C
      NEL = NE(J,N,NANGEL,JE)
C
C LOOP OVER THE ANGULAR ELEMENTS CONTAINING GLOBAL NODE J
C AND CALCULATE THE CONTRIBUTION EACH MAKES TO THE INTEGRAL.
C
      DO 2 K=1,NEL
C
C     DETERMINE THE ELEMENT NUMBER OF THE K'TH ANGULAR ELEM.
C
      KEL = KE(J,K,N,NANGEL,JE)
C
C     DETERMINE THE LOCAL NODE NUMBER OF THE GLOBAL NODE J
C     IN THE K'TH ANGULAR ELEMENT CONTAINING IT.
C
      LOCAL = LNN(J,KEL,N,JE)
C
C     THE VALUE LOCAL MUST BE PASSED TO FUNCTION TRIAL2(SEE
C     COMMON/LNODE).
C
C     SET UP THE CANONICAL MAPPING OF THE K'TH ANGULAR
C     ELEMENT TO BE INTEGRATED.
C
```

```
      CALL MAP2(ALO(KEL),AHI(KEL),AFAC,CLO(KEL),CHIHI(KEL),
     $   CFAC)
      FAC = AFAC * CFAC
C
C     CARRY OUT NUMERICAL QUADRITURE OF THE K'TH ANGULAR EL.
C
      GAUSS = 0.
      DO 1 I=1,IPOINT
      XI = Q(I)
      AMUI = AMU2(XI)
      DO 1 J=1,IPOINT
      ETA = Q(J)
      CHIJ = CHI2(ETA)
      GAUSS = GAUSS + W(I)*W(J)*FCT1(R,Z,AMU,CHI,AMUI,CHIJ)
     $   *FCT2(XIK,ETAL,XI,ETA)*FAC
    1 CONTINUE
      GAUSS2 = GAUSS2 + GAUSS
    2 CONTINUE
      RETURN
      END

      FUNCTION NE(J,N,NANGEL,IE)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
C THIS SUBROUTINE CALCULATES THE NUMBER OF ELEMENTS CONTAIN*
C ING GLOBAL NODE J IN THE ANGULAR DOMAIN.                        *
C * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
C
      DIMENSION IE(50,16)
      INTEGER E
      I = 0
      DO 2 E=1,NANGEL
      DO 1 K=1,N
      IF(J.EQ.IE(E,K)) I=I+1
      IF(I.EQ.4)GO TO 3
    1 CONTINUE
    2 CONTINUE
    3 NE = I
      RETURN
      END

      FUNCTION LNN(J,KEL,N,IE)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
C THIS SUBROUTINE CALCULATES THE LOCAL NODE NUMBER OF GLOB-*
C AL NODE J IN ELEMENT KEL.                                       *
C * * * * * * * * * * * * * * * * * * * * * * * * * * *   * * * * *
C
      DIMENSION IE(50,16)
      DO 1 I=1,N
      IF(IE(KEL,I).EQ.J)GO TO 2
    1 CONTINUE
    2 LNN = I
      RETURN
      END
      FUNCTION KE(J,K,N,NANGEL,IE)
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THIS SUBROUTINE CALCULATES THE NUMBER OF THE K'TH ANGU-         *
C LAR ELEMENT CONTAINING GLOBAL NODE J.                           *
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DIMENSION IE(50,15)
      INTEGER E
      KK = 0
      DO 2 E=1,NANGEL
      DO 1 I=1,N
      IF(J.EQ.IE(E,I))KK=KK+1
      IF(KK.EQ.K)GO TO 3
    1 CONTINUE
    2 CONTINUE
    3 KE = E
      RETURN
      END
```

Appendix B

## Derivation of the Weak Form of the Transport Equation

From Chapter II we have the functional

$$F(u) = \int_D \Big[ \overset{\textcircled{1}}{\langle \underline{\Omega}\cdot\nabla u, K_u(\underline{\Omega}\cdot\nabla u)\rangle} + \overset{\textcircled{2}}{\langle u, G_g u\rangle}$$

$$- 2\overset{\textcircled{3}}{\langle \underline{\Omega}\cdot\nabla u, K_u S_u\rangle} - 2\overset{\textcircled{4}}{\langle u, S_g\rangle} \Big] d\underline{r} + \oint_R \Big[ \overset{\textcircled{5}}{\int |\underline{\Omega}\cdot\underline{n}(\underline{r})|\, u^2 \, d\underline{\Omega}} \Big] d\underline{r} \qquad (23)$$

where $\langle f,g\rangle = \int_{\underline{\Omega}} f^*(\underline{\Omega})\, g(\underline{\Omega})\, d\underline{\Omega}$ and * stands for the complex conjugate. We want to show that $F(u) > F(\psi)$ where $u = \psi + \eta$ and $\eta \neq 0$. From Eq (23) above, we have

$$F\ \psi+\eta = \int_D \langle(\underline{\Omega}\cdot\nabla\psi), K_u(\underline{\Omega}\cdot\nabla\psi)\rangle\, d\underline{r} \qquad \text{(B-1)}$$

$$+ \int_D \langle(\underline{\Omega}\cdot\nabla\psi), K_u(\underline{\Omega}\cdot\nabla\eta)\rangle\, d\underline{r} \qquad \text{(B-2)}$$

$$+ \int_D \langle(\underline{\Omega}\cdot\nabla\eta), K_u(\underline{\Omega}\cdot\nabla\psi)\rangle\, d\underline{r} \qquad \text{(B-3)}$$

$$+ \int_D \langle(\underline{\Omega}\cdot\nabla\eta), K_u(\underline{\Omega}\cdot\nabla\eta)\rangle\, d\underline{r} \qquad \text{(B-4)}$$

$$+ \int_D \langle \psi, G_g \psi\rangle\, d\underline{r} \qquad \text{(B-5)}$$

$$+ \int_D \langle \psi, G_g \eta\rangle\, d\underline{r} \qquad \text{(B-6)}$$

$$+ \int_D \langle \eta, G_g \psi\rangle\, d\underline{r} \qquad \text{(B-7)}$$

$$+\int_D \langle \eta, G_g \eta \rangle \, d\underline{r} \qquad \text{(B-8)}$$

$$-2\int_D \langle \underline{\Omega}\cdot\nabla\psi, K_u S_u \rangle \, d\underline{r} \qquad \text{(B-9)}$$

$$-2\int_D \langle \underline{\Omega}\cdot\nabla\eta, K_u S_u \rangle \, d\underline{r} \qquad \text{(B-10)}$$

$$-2\int_D \langle \psi, S_g \rangle \, d\underline{r} \qquad \text{(B-11)}$$

$$-2\int_D \langle \eta, S_g \rangle \, d\underline{r} \qquad \text{(B-12)}$$

$$+\int_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}(\underline{r})| \, \psi^2 \, d\underline{\Omega} \, d\underline{r} \qquad \text{(B-13)}$$

$$+2\int_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}(\underline{r})| \, \psi \, \eta \, d\underline{\Omega} \, d\underline{r} \qquad \text{(B-14)}$$

$$+\int_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}(\underline{r})| \, \eta^2 \, d\underline{\Omega} \, d\underline{r} \qquad \text{(B-15)}$$

It can be seen that Eq (B-1) + Eq (B-5) + Eq (B-9) + Eq (B-11) + Eq (B-13) equal $F(\psi)$. Since the operators $G_g$, $G_u$, $K_g$, and $K_u$ are <u>self-adjoint</u>, that is,

$$\langle f(\underline{\Omega}), G_g\, g(\underline{\Omega}) \rangle = \langle G_g\, f(\underline{\Omega}), g(\underline{\Omega}) \rangle$$

$$= \langle g(\underline{\Omega}), G_g\, f(\underline{\Omega}) \rangle^*$$

we have terms

(B-2) = (B-3)*

and

(B-6) = (B-7)*

Therefore, terms

$$(B\text{-}2) + (B\text{-}3) = (B\text{-}3)^* + (B\text{-}3) = 2\mathrm{Re}(B\text{-}3)$$

and we find that

$$F(\psi,\eta) = F(\psi) \tag{A}$$

$$+\, 2\Bigg[\int_D \Big[\underbrace{\mathrm{Re}\langle \underline{\Omega}\cdot\nabla\eta, K_u(\underline{\Omega}\cdot\nabla\psi)\rangle}_{(B-3)} + \underbrace{\mathrm{Re}\langle \eta, G_g\psi\rangle}_{(B-7)}$$

$$- \underbrace{\langle \underline{\Omega}\cdot\nabla\eta, K_u S_u\rangle}_{(B-10)} - \underbrace{\langle \eta, S_g\rangle}_{(B-12)}\Big]\, d\underline{r}$$

$$+ \underbrace{\oint_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}(\underline{r})|\,\psi\eta\; d\underline{\Omega}\, d\underline{r}}_{(B-14)}\Bigg] \tag{B}$$

$$+ \Bigg[\int_D \Big[\underbrace{\langle \underline{\Omega}\cdot\nabla\eta, K_u(\underline{\Omega}\cdot\nabla\eta)\rangle}_{(B-4)} + \underbrace{\langle \eta, G_g\eta\rangle}_{(B-8)}\Big]\, d\underline{r}$$

$$+ \underbrace{\oint_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}(\underline{r})|\,\eta^2\; d\underline{\Omega}\, d\underline{r}}_{(B-15)}\Bigg] \tag{C}$$

Because the operators $G_g$, $G_u$, $K_g$, and $K_u$ are all positive-definite, the terms (B-4), (B-8), and (B-15) in the bracketed (C) term are positive as long as $\eta$ is not equal to zero. Thus, the (C) term is positive.

Similarly, the operators $G_g$, $G_u$, $K_g$, and $K_u$ are real operators; they give real results when operating on real functions. So, as long as we use real functions $\psi$ and $\eta$ , we can drop the two Re's in the bracketed term (B).

Using the property

$$\int_D \langle \underline{\Omega}\cdot\nabla\eta, f\rangle \, d\underline{r} = -\int_D \langle \eta, \underline{\Omega}\cdot\nabla f\rangle \, d\underline{r}$$

$$+ \oint_R \langle (\underline{\Omega}\cdot\underline{n}(\underline{r}))\eta, f\rangle \, d\underline{r} \qquad \text{(B-16)}$$

the (B-3) and (B-10) terms in the bracketed term (B) can be written as

$$\int_D \langle \eta, -\underline{\Omega}\cdot\nabla\Big[K_u[\underline{\Omega}\cdot\nabla\psi]\Big]\rangle \, d\underline{r}$$

$$+ \oint_R \langle (\underline{\Omega}\cdot\underline{n})\eta, K_u(\underline{\Omega}\cdot\nabla\psi)\rangle \, d\underline{r} \qquad \text{(B-3)}'$$

and

$$\int_D \langle \eta, -\underline{\Omega}\cdot\nabla(K_u S_u)\rangle \, d\underline{r}$$

$$+ \oint_R \langle (\underline{\Omega}\cdot\underline{n})\eta, K_u S_u\rangle \, d\underline{r} \qquad \text{(B-10)}'$$

Bracketed term (B) thus becomes

$$2\int_D \Big[ \langle \eta, -\underline{\Omega}\cdot\nabla[K_u(\underline{\Omega}\cdot\nabla\psi)]\rangle + \langle \eta, G_g\psi\rangle$$

$$+ \langle \eta, \underline{\Omega}\cdot\nabla(K_u S_u)\rangle - \langle \eta, S_g\rangle \Big] \, d\underline{r}$$

$$+ 2\oint_R \Big[ \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}|\psi\eta \, d\underline{\Omega} + \langle (\underline{\Omega}\cdot\underline{n})\eta, K_u(\underline{\Omega}\cdot\nabla\psi)\rangle$$

$$- \langle (\underline{\Omega}\cdot\nabla\underline{n})\eta, K_u S_u\rangle \Big] \, d\underline{r}$$

$$= 2\int_D \Big[ \Big< \eta, -\underline{\Omega}\cdot\nabla[K_u(\underline{\Omega}\cdot\nabla\psi)] + G_g\psi - [S_g - \underline{\Omega}\cdot\nabla(K_u S_u)] \Big> \Big] d\underline{r}$$

$$+ 2\oint_R \int_{\underline{\Omega}} \eta \Big[ |\underline{\Omega}\cdot\underline{n}|\psi + (\underline{\Omega}\cdot\underline{n})[K_u(\underline{\Omega}\cdot\nabla\psi) - K_u S_u] \Big] d\underline{\Omega}\, d\underline{r}$$

Eqs (17), (18a), and (18b) of Chapter II are now applied to force term (B) to zero. Thus, $F(\psi+\eta)$ is indeed greater than $F(\psi)$. The resulting <u>weak form</u> of the transport equation corresponding to its variational formulation, Eq (23), is

$$\int_D [ < \underline{\Omega}\cdot\nabla\eta, K_u(\underline{\Omega}\cdot\nabla\psi) > + < \eta, G_g\psi > ] \, d\underline{r}$$

$$+ \oint_R \int_{\underline{\Omega}} |\underline{\Omega}\cdot\underline{n}| \eta\psi \, d\underline{\Omega}\, d\underline{r}$$

$$= \int_D [ < \underline{\Omega}\cdot\nabla\eta, K_u S_u > + < \eta, S_g > ] \, d\underline{r} \qquad (24)$$

## VITA

Ronald Clark Wheaton was born on 13 March 1946 in Steubenville, Ohio. He graduated from high school in Wintersville, Ohio in 1964 and attended Ohio University from which he received the degree of Bachelor of Science in June 1968. After completing Officer Training School in June 1969, he was commissioned in the USAF. Upon completion of navigator training in April 1970, he served as a C-133 navigator at Travis AFB, California and as a C-130 navigator at CCKAB, Taiwan. He also served as a C-141 navigator at Charleston AFB, South Carolina until entering the School of Engineering, Air Force Institute of Technology, in June 1977.

Permanent address: Route 5, Two Ridge Road
Wintersville, Ohio 43952